**A Project Report On**

**Capture It**

**Submitted in Partial Fulfillment of Requirement**

**For the Award of the Degree**

**BACHELOR OF COMPUTER APPLICATION**

**MANGALORE UNIVERSITY**



**Under the guidance of**

**Asst. Prof. Sushma**

**Department of Computer Science**

**St. Mary’s College, Shirva**

**By**

**Roland Mendonca 173925019**



**DEPARTMENT OF COMPUTER SCIENCE**

**ST.MARY’S COLLEGE SHIRVA-5741162019-2020**



**Department Of Computer Science**

**CERTIFICATE**

This is to certify that the project work entitled **“Capture It”** has been carried out by Roland Mendonca (173925019) students of VI semester BCA, in partial fulfillment of the requirements to the completion of VI semester, Bachelor of Computer Applications of Mangalore University during the year 2019-2020.

**Internal Guide**

Asst. Prof. Sushma

Department of Computer Science

**Head of the Department Principal**

Prof. K. Praveen Kumar

Dept. of Computer Science

**Examiners: Date:**

**1.**

**2.**

**ACKNOWLEDGEMENT**

The project is like when you put your passion and dedication on a single piece of work by enhancing our knowledge and opening our minds for the technology and it’s an immense happiness when the project that we have worked so hard to build comes out to be the way we have always thought it would be.

Making such a complex project in a limited time is a challenge in itself. So, there are a lot of people who always have our back when we run out into problems. Those people lift us up and we couldn’t have imagined our project turning out so goo without them.

First of all, we would like to thank them almighty god for showering his abundant blessing on us. And I’m grateful to our Principal Prof. Herald Ivan Monis for supporting us throughout the project development time and providing us to use all the facilities of the college needed for the environment.

I would like to thank Prof. K Praveen Kumar, Department of Computer Science St. Mary’s College Shirva, for all the concern and support extended throughout the project.

I express my heart full thank to Mr. Prakash, Ms. Sushma and Ms. Divyashree, lectures of Department of Computer Science St. Mary’s College Shirva, for their encouragement and support extended in completing the project.

Finally, the word of thanks goes to my family and friends for guiding encouraging and inspiring me all through the project.

Thank you all,

Date: **Roland Mendonca**

Place:

**Declaration**

I hereby declare that our project entitled “**Capture It**” by us for the partial fulfillment for the award of degree of Bachelor in Computer Applications, of Mangalore University embodies the result of our project work carried out under the direct supervision and guidance of Asst. Prof Divyashree, our internal guide, and the support of Ass. Prof. K. Praveen Kumar, Head of the Department of Computer Science, during the period of December 2018 to March 2019. We further declare that the part of this project has not been previously submitted to any other examination or institution.

Date: **Roland mendonca**

Place:

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Contents** | **Page No** |
| **1** | **Synopsis** | **1-3** |
|  | * 1. Title of the project   2. Project category   3. Objective of the project   4. Structure of the project   5. Scope of the project   6. Modules   7. Modules description   8. Hardware and Software requirement |  |
| **2** | **Literature survey** | **5-8** |
|  | 2.1 PHP  2.2 MySql  2.3 CSS  2.4 HTML5  2.5 AJAX  2.6 Java script  2.7 RDBMS |  |
| **3** | **Software requirement Specification** | **10-13** |
|  | 3.1 Introduction  3.2 Purpose  3.3 Overview  3.4 Overall Description  3.5 Product Perspective  3.6 Product Feature  3.7 Definition, Acronyms and Abbreviation  3.8 User Classes and User Characteristics  3.9 Design and Implementation constrains  3.10 Assumption and Dependencies  3.11 Specific Requirements  3.12 Programming language  3.13 Communication Interface  3.14 Functional Requirements  3.15 Documents Convention  3.16 System Feature  3.17 Other non-functional requirements |  |

|  |  |  |
| --- | --- | --- |
| **4** | **System Design** | **15-21** |
|  | 4.1 Introduction  4.2 Context Flow Diagram  4.3 Data Flow Diagram  4.4 DFD Notations   * Process * Data Flow * External Entity   4.5 Rules Constructing DFD |  |
| **5** | **Database Design** | **23-32** |
|  | 5.1 Introduction  5.2 schema description  5.3 Entity-Relationship(ER) Diagram  5.4 ER Notations  5.5 ER Diagram |  |
| **6** | **Detail Design** | **34-40** |
|  | 6.1 Introduction  6.2 Structured English |  |
| **7** | **System Coding** | **42-69** |
|  | 7.1 Introduction  7.2 Module Coding |  |
| **8** | **System Testing** | **71-74** |
|  | 8.1 Introduction  8.2 Psychology of testing  8.3 Levels of testing |  |
| **9** | **Snapshots** | **76-94** |
| **10** | **Future Scope of the Project** | **96** |
| **11** | **Conclusion And Future Enhancement** | **98** |
| **12** | **Limitations** | **100** |
| **13** | **Bibliography** | **102** |

**Chapter-1**

**SYNOPSIS**

**1.1 Title of the Project**

Capture It

**1.2 Main objective of the project**

* To Navigate Through our Work.
* To book service.
* Admin has user interface to manage his work, service and bookings.

**1.3Hardware requirements**

Processor – I3 or more

Hard disk – 3GB or more

RAM - 1GB or more

Internet connectivity

**1.4 Software requirements**

Windows 7 or higher.

**1.5 Languages used**

**Front end :-**

PHP, HTML, CSS, JS

**Back end :-**

MySQL

**1.6 Structure of project**

* To decrease Time and cost in Adding or Changing the work or Services.
* Booking Services.
* Easy navigation of work.

**1.7 Features**

* Acts as bridge between the Service Provider and customer.
* Helps user to naviagate through work or service easily.
* Helps Admin to Adding or Changing the work or Services easily.

**1.8 Modules**

**Admin**

* **Login/Register**

To create account or to login as your permanent account. It is used to maintain admin data.

* **Profile**

Contains admin profile data.

* **Category**

Gives the category on Photography, videography, aerial, 360, VR, Editing.

* **Services**

Can add or remove various services.

* Query

Receive the user query and answer them.

* **report**

Gives the in Between date report to admin.

* **Booking**

Gets request from the client on booking. Gives the choice to accept or reject, He can also Edit his choice.

* **Notification**

User will get email Notifications.

**User**

* **Login/Register**

To create account or to login as your permanent account. It is used to maintain Client data.

* **Category**

Go through category and Navigate works.

* **Sending Booking request**

Send the request to the Service provider for his service.

* **Query**

Can send his Query.

**Chapter-2**

**LITERATURE SURVEY**

2.1 PHP



* PHP Stands for Hypertext Preprocessor, which is a recursive acronym.
* PHP is a server side scripting language designed for web development but also used as a general purpose programming language.As of January 2013,PHP was installed on more than 240 million websites and 2.1 million web servers.

**Some features of PHP:**

* Very simple and easy to use - comparing to other scripting language.
* Interpreted language-no need for compilation.
* Faster than other scripting language-Eg : asp and jsp.
* Open source-no need to pay for use PHP .
* Platform Independent-PHP code can run on any platform, Linux, Unix.

Mac OS Windows

Case sensitive scripting language at time of variable declaration.

2.2 My SQL



MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company.

MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.

2.3 CSS (cascading style sheets)



CSS is a style sheet language used to describe the presentation semantics (that is, the look and formatting) of a document written in a markup language (Style defines how to display HTML elements). Its most common application is to style web page written in HTML and any kind of XML documents. Cascading Style Sheets (CSS) are widely recognized for their contributions in building fact-loading, standards compliant, easily modifiable web pages. External Style Sheets are stored in CSS files CSS3 is the latest standard for CSS. CSS3 is completely backwards-compatible with earlier versions of CSS. The Feedback Engine Project uses the CSS3 specifications.

2.4 HTML5



* HTML stands for Hyper Text Mark-up Language.
* A mark-up language is the set of mark-up languages.
* A HTML Mark-up tags are usually called as HTML Tags.
* HTML tags are the keywords surrounded by angle brackets like <html>.
* HTML document contain HTML tags and plain text.
* HTML documents are also called as the web pages.
* PHP code can be simply mixed with HTML code is usually proceeded by a PHP interpreter, which is usually implemented as a web servers native module or a common gateway Interface (cgi) executable.
* PHP files can contain text, HTML, tags and scripts.PHP files can have a file extensions of php, petal,.php3,.php4,.php5,or PHS.

2.5 JAVA SCRIPT



* JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.
* JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages .It is a multi paradigm language, supporting object-oriented, imperative, and functional programming styles.

2.6 XAMPP



XAMPP is a free and open source cross platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP server, Maria DB Database, and interpreters for scripts Written in the PHP and pearl programming Languages.XAMPP stands for Cross-platform(X),Apache(A),Maria DB(M),PHP(P) and pearl(P).it is simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server-server application (Apache), database(Maria DB),and scripting Language(PHP)-is included in an extractable file.XAMPP is also cross-platform, which means it works equally well on Linux.Mac and windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to live server extremely easy as well.

**CHAPTER-3**

**SOFTWARE SPECIFICATIONS AND REQUIREMENTS**

3.1 Introduction

A software requirement specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of used cases that describe interactions the users will have with the software.

A basic purpose of the SRS is to bridge the communication gap between client and the developer so they have a shared vision of the software being built. An SRS establishes the basic for agreement between the client and the supplier on what the software product will do. SRS provides a reference for validation of the final product. A high-quality SRS is a prerequisite to high-quality software and also reduces the development cost.

The introduction of the software requirement specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim this document is to gather and analyze and give an in-depth insight of the complete “Capture It” by defining the problem statement in detail. The detailed requirements of “Capture It” are provided in the document.

3.2 Purpose

This document includes software requirements for the “Capture It” project. The purpose of this document is to detail the user requirements for all the functionality “Capture It”. This document is meant to serve as a guide to the developers and users. The purposes of this requirement document are to specify and provide all the information required to design, develop and test the system. This document ensures that the person reading the document understands what he/she is looking for.

The main purpose of this “Capture It” is to provide an user friendly website where they can select the best required services for their event. And also allow them to ask queries or give Reviews.

3.3 Overview

This SRS will allow for a complete understanding of what is to be expected of the “Capture It” Project and its functionality will allow for end users and will be used for the development of the future stages of the project.

3.4 Definitions, Acronyms and abbreviations

SRS – Software requirement system

OS – Operating system

PHP – PHP: Hypertext Preprocessor

PC – Personal Computer

MB – Mega bytes

RAM – Random access memory

MySQL – MySQL database Server

3.5 Overall Description

An overall description of our project is to make the customer to easy to book and Service provider easy to Maintain.

The customer has the option to , book service and ask question.

The admin has option to manage service, manage work, answer query's of customer and manage bookings.

3.5.1 Product perspective

For an Outskirt customer needs an application to Navigate work, choose service or ask query's. This application will make it easy to choose his required service.

3.5.2 Product feature

This is a web project with frontend with PHP and Back end with MySql.

* Maintain an admin side to make them List the Services Provided with them.
* Maintain an admin side to Easily Add or Update their work.
* Maintain an admin side to Accept or Reject the customer request.
* Maintain an user side to book the services or to ask queries .
* The Request for booking by the customer is sent to admin, and admin will contact the customer by other means of communication.
* User has the option to ask the queries.
* User must create his account before sending a Booking request. This will help us in keeping the records which can be used to give reports.

3.5.3 User’s classes and characteristics:

Admin:

For the first time the admin has to register than in future he can Login and edit the info he has given.

Customer:

The customer can view the services which are listed on the website But to send the request for booking he has to login/register this will help us to keep track on their previous activity.

3.5.4 General constraints

* This project is based on web.
* The Services can be listed by the service provider.
* The customer can see all the Listed services and send the booking requests to the service provider.
* For sending the booking request the customer has to register/login.

3.5.5 Assumptions and dependencies:

* Internet connection required.
* The user should know the basic functioning on Web.
* One assumption about the Project is that it will always be used on device that has enough performance.

3.6 External interface requirements:

This section provides a detailed description of  all inputs into and output from the system. It also gives a description of the hardware software and communication interfaces and provides basic prototypes of the user interface.

3.6.1 User Interface:

  User interface is a graphical user interface consisting of Tab forms and bottom-navigation menus. The user interface must be simple and easy to use. All interaction of the software with the people must be easy and clearly specified. This is a project provides UI and command buttons, text boxes, labels, and data grids are used.

3.6.2 Hardware and Software interfaces:

 Browser: Any latest browser preferably chrome

3.7 Functional requirements:

  The functions of the form used in the application are as follows it specifies which output should be produced from the given input and they describe the relation between inputs and outputs of the system. All the operation performed on the input   data to obtain the specified output should be specified. The application has two characteristics admin/service provider and customer.

 Admin/Service provider owner has following privileges.

* Create his account.
* Edit his profile.
* Add or remove the service.
* Check bookings.
* Check Query's.
* Check report between dates.

 Customer has following privileges.

* View service information.
* Edit or create their account information.
* Request for service.
* Send his Query's.
* Check his previous bookings.

3.8 Non functional requirements:

* Performance requirements

  (This section describes the performance requirements expected from this project)

* Quick navigation between pages.
* Various validations of a user inputs.
* Email notification.
* Easy to use.
* Security requirements

(section describes the security requirements expected from this project)

* The every user should login to the system to manipulate book any services.
* The system shall permit only service provider has the permission to edit his profile.
* Email notification.

**SYSTEM DESIGN**

**CHAPTER-4**

**System design**

4.1 INTRODUCTION

Problem specified by the requirement document. this phase moves from the problem domain to the solution domain. Design is a process through which requirements are translated into a representation of the software. The purpose of the design phase is to plan a solution for the problem specified by the requirement document. This phase moves from the problem domain to the solution domain.

1. Architecture design

2. High level design

3. Detailed design

In an architectural design the focus is on identifying the components or subsystems and how they interact to each other. The high level design identifies the modules that should build for developing the system. Incase of detailed design the focus is on how the modules are implemented in software.

System design is a software,a ‘how to approach to the creation of a new system.The important phase is composed of several steps. it provides the understanding of procedural details necessary for the implementation the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specifications.

4.2 Methodology/objective

The main objectives of the design are:

1. Practicality

2. Efficiency

3. Flexibility

4. Completeness

5. Security

6. Verifiability

7. Traceability

**4.3 Design Methodology:**

BS carried out the project using a module driven development.By the input obtained from the customer,analysis of the system and feasibility study,a model of the system is developed and further proceedings are based on the initial model.Changes have been made to the model in frequent intervals,according to need using some standards and the final system developed.

**4.4 Context Flow diagram(CFD):**

This diagram pictures the system at the center,with no details of its interior structure,surrounding by all its interacting systems and activities.Main objective is to focus on external factors and events.

**4.5 Data Flow Diagrams(DFD):**

The data flow diagram is way of expressing system requirements in a graphical form. A DFD also known a bubble chart as purpose of clarifying system requirements and identifying major transformations that will become programs in design.

**4.6 DFD Notations**

Process

Data Flow

Data store

External Entity

**Process:**

  A circle or a bubble represents a process that transforms data from one from to another by performing some tasks with the data the process name must be given a general idea of its function

**Data flow:**

  An arrow represents data flow .it represents the path over which travels in system.A data flow can move between processes, flow into or out of data stores,to and form external entities it must be given a name above the arrowhead showing the direction of flow.

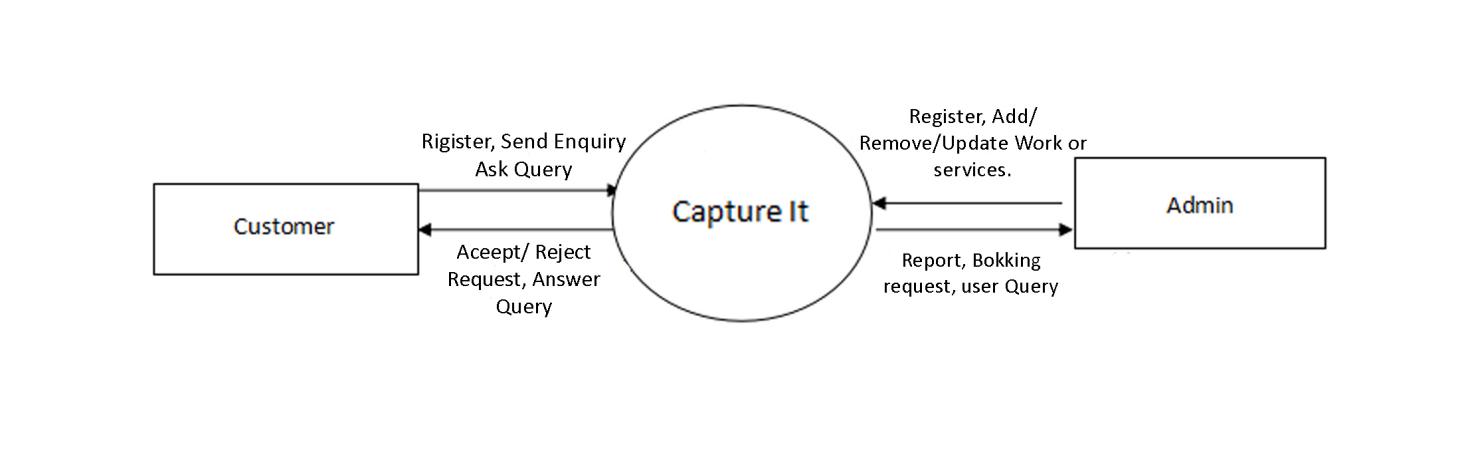
**External Entity**

 A rectangle, which defines the source or destination of system data also called as external entity. An external entity is not possible for any task performed by the system.

**4.7 Rules for constructing DFD**

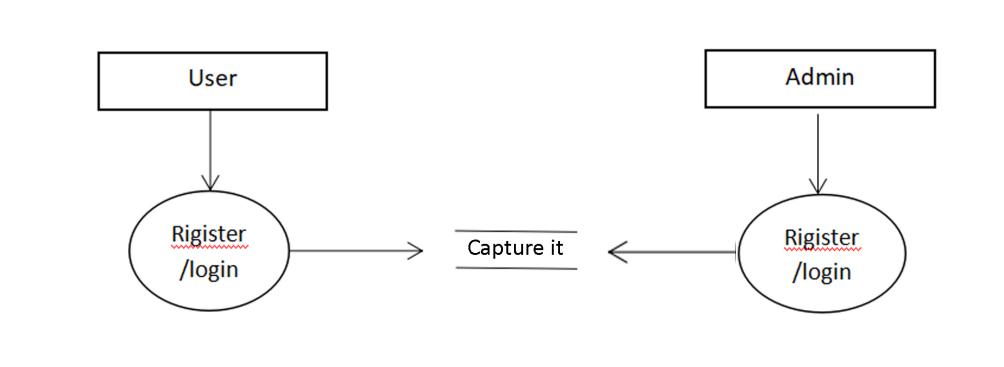
1. Process should be named for easy understanding.
2. The direction of flow,top to bottom and form left to right should be specified.
3. The direction flow should not allow any kind of loops.

**Context flow diagram**

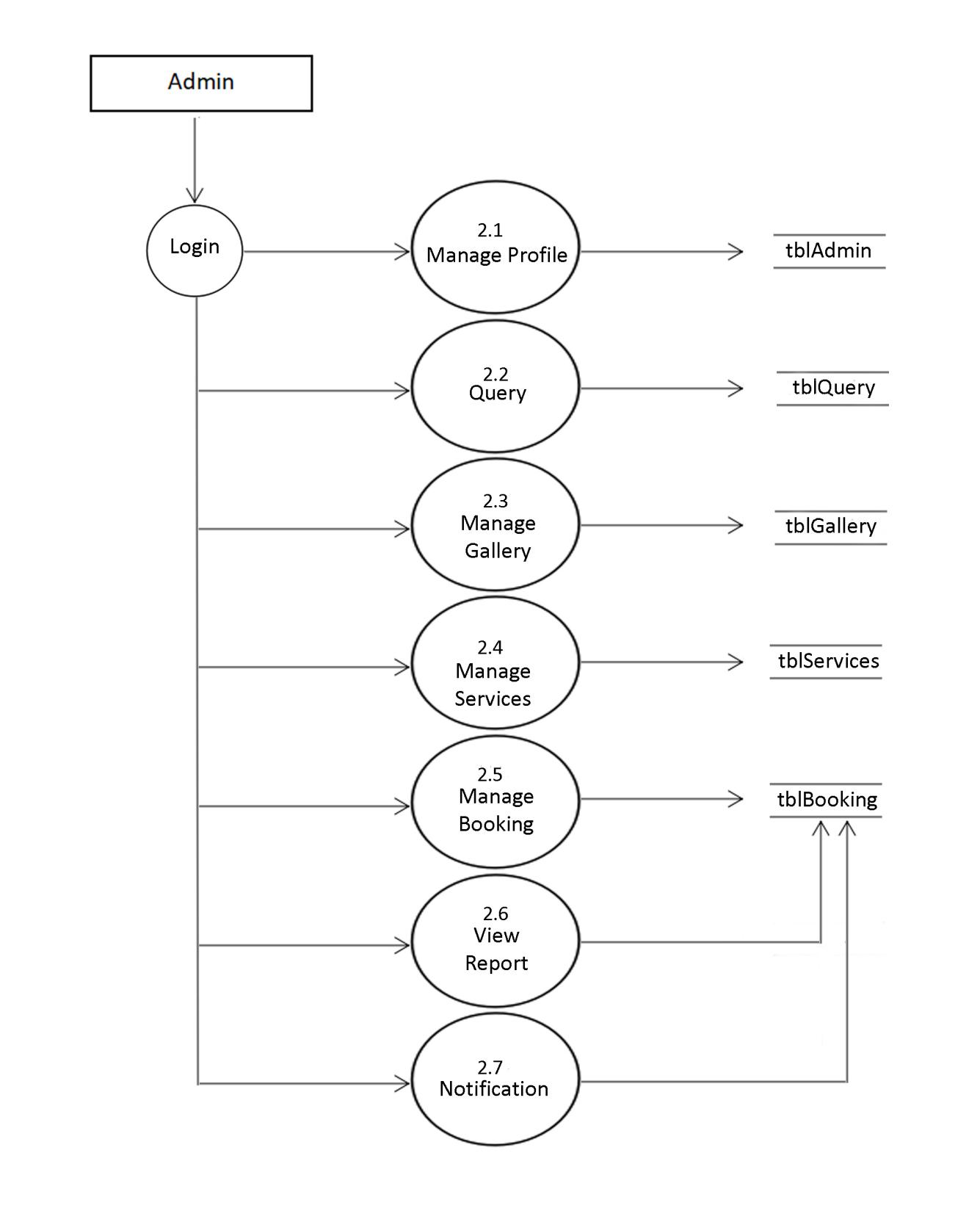


v

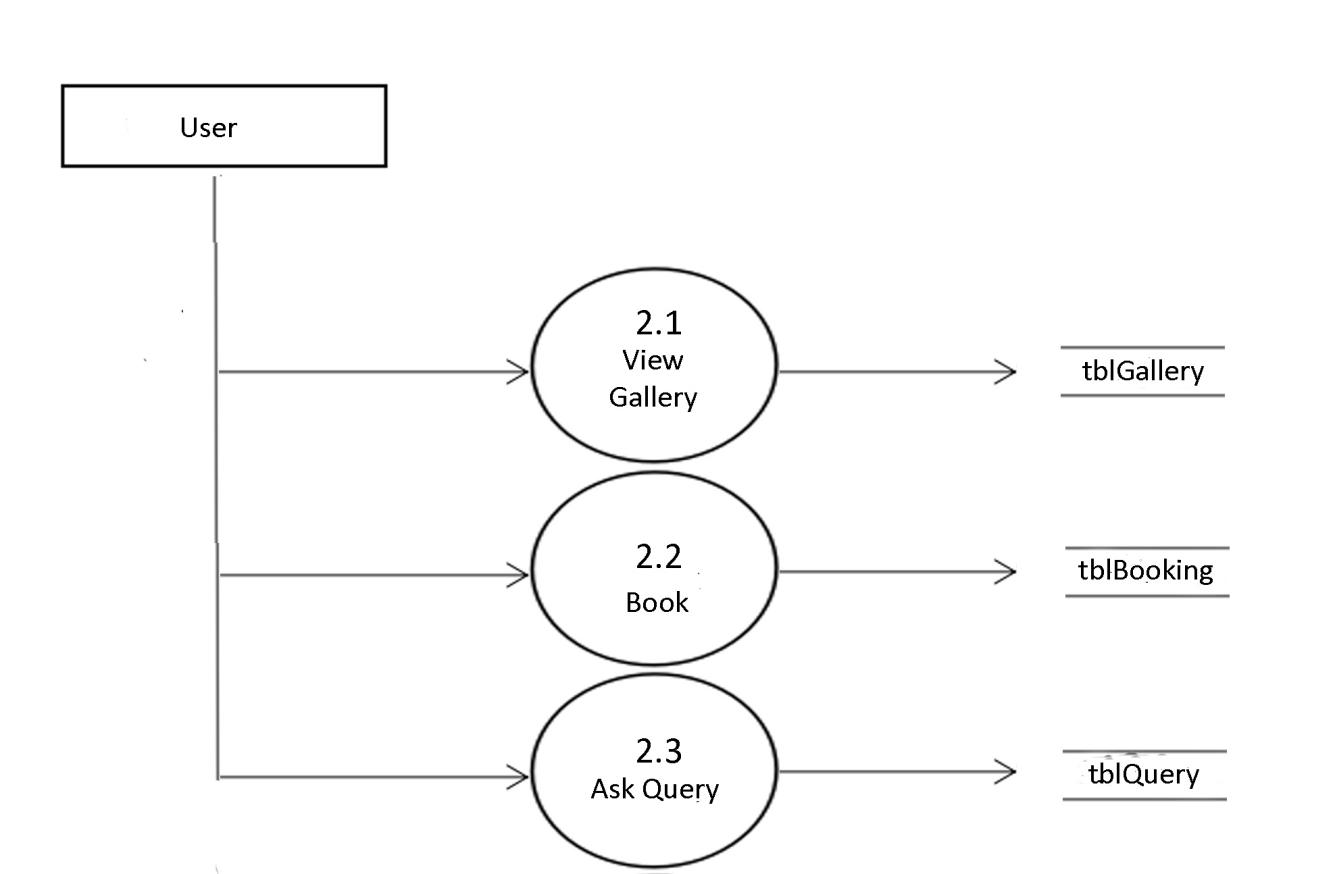
**DFD level 1**



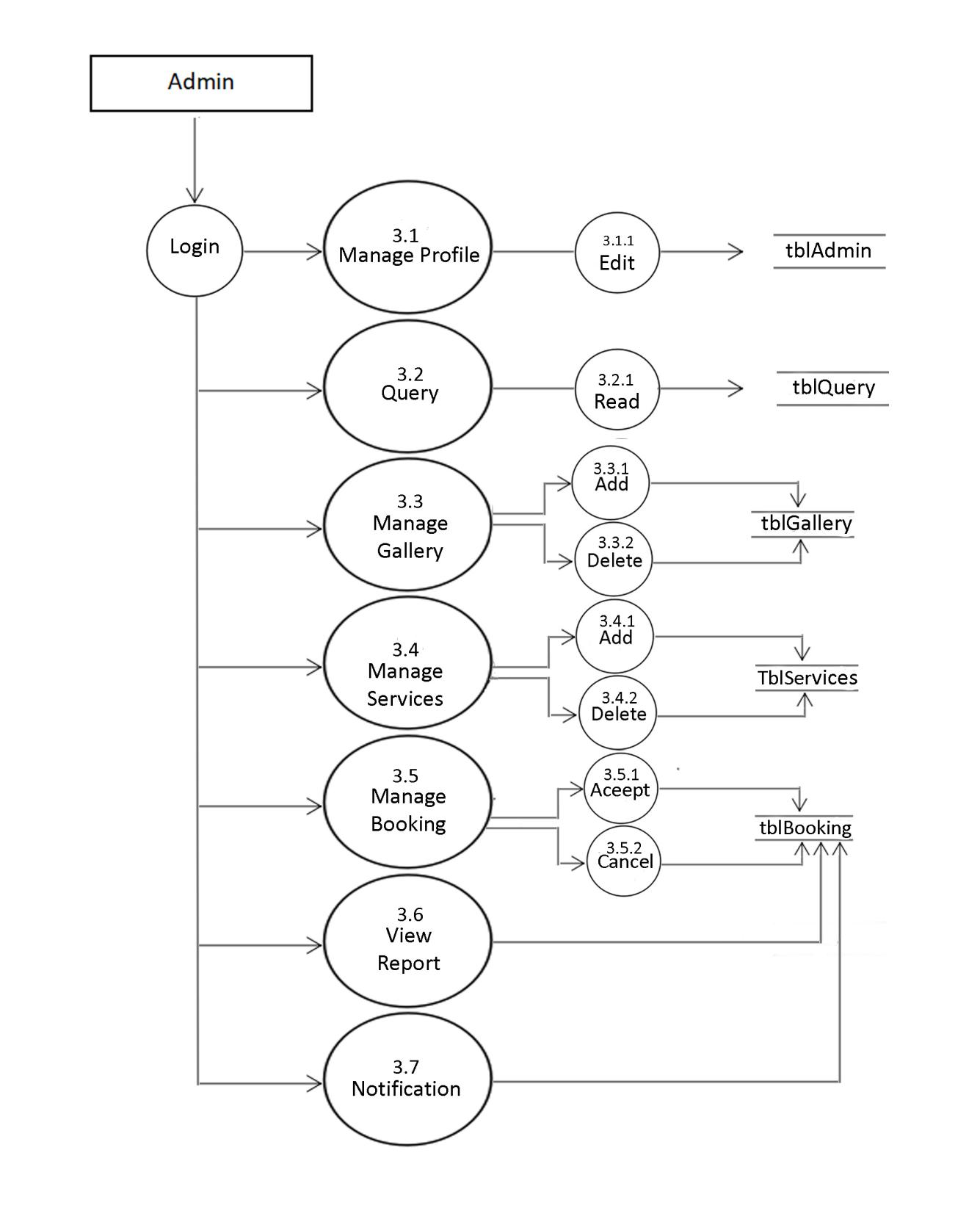
**DFD level 2(admin)**



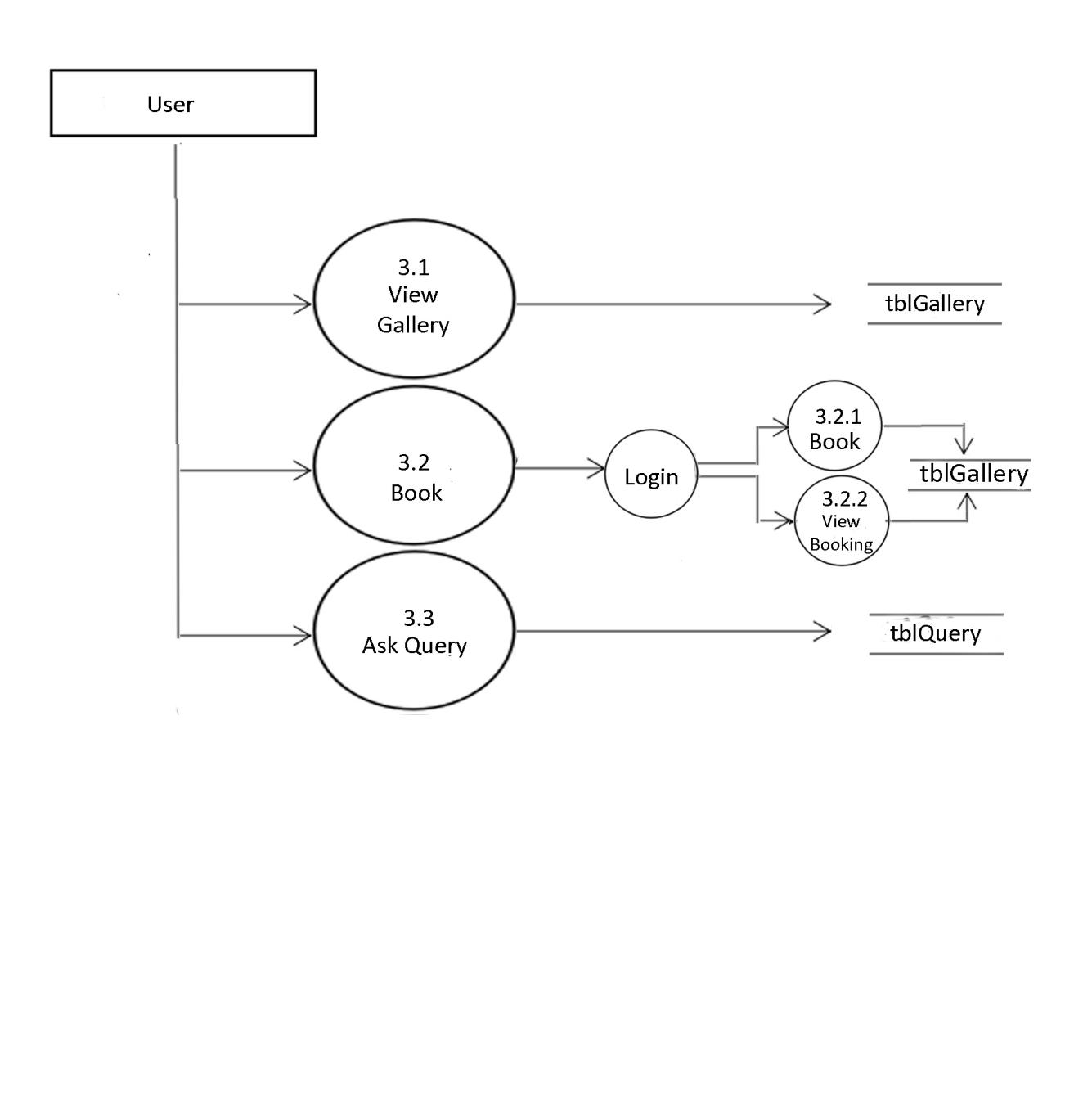
**DFD level 2(user)**



**DFD level 3 (admin)**

****

**DFD level 3(user)**



**DETAILED DESIGN**

**Chapter 5  
DATABASE DESIGN**

**5.1 Introduction:**

A database is an inherent collection of data with some inherent meaning designed, built populated with data for specific purpose.the following guideline is been following during the database design.

* Descriptive names for the tables,columns and indexes.
* Singular names for tables and columns.
* Proper data type each column.

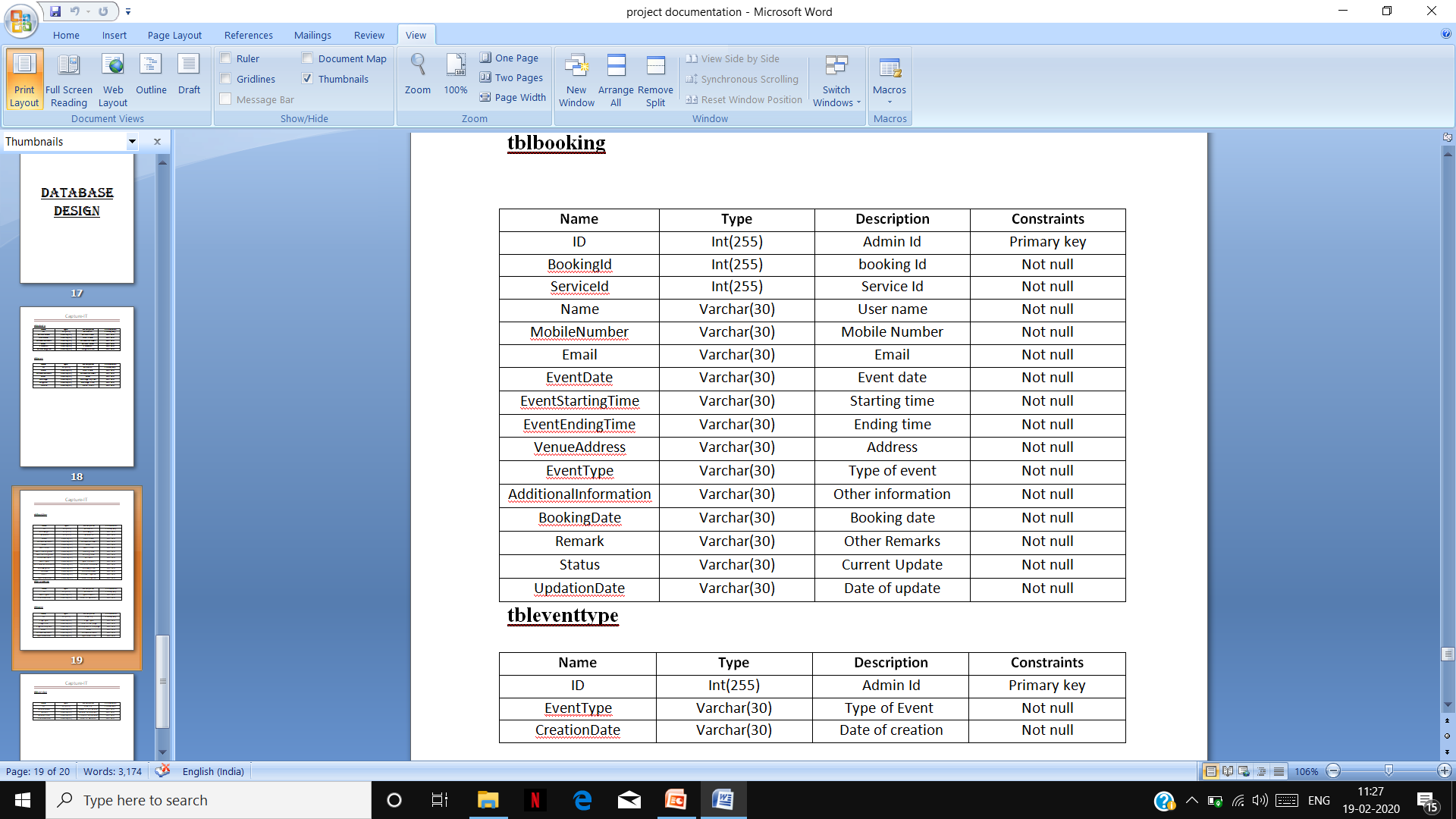
**5.2 Database:**

A database is collection of related data,which can be of any size and complexity.By using concept of database,we can easily store and retrieve the data.The major purpose of database is to provide the information,which utilizes it with the information’s that the system needs according to its own requirements.

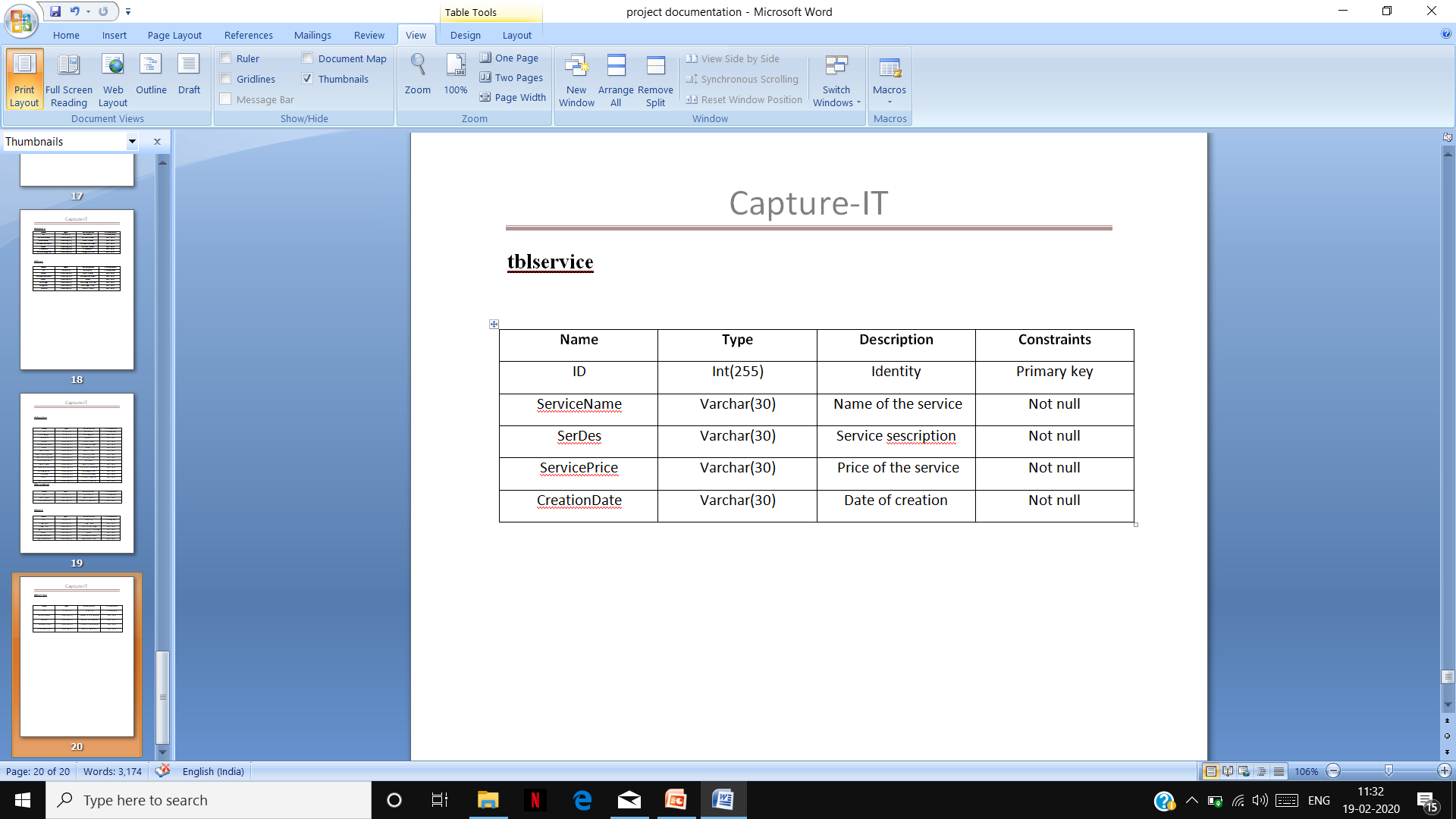
**5.3 Database Design**

Database Design is done before building it to meet need if the end user within a given information system that the database is intended to support.The database design defines The needed data and the data structures that a database comprises.The database is physically implemented using my Sql.



****

****

****

**tbl360Categories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| Cv\_no | Int(255) | Video Category number | Primary key |
| Category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblaerialcategories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| Ca\_no | Int(255) | Aerial Category number | Primary key |
| Category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblaerialgallery**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| ga\_no | Int(255) | Aerial gallery number | Primary key |
| Category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tbleditingcategories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| Cv\_no | Int(255) | Video editing number | Primary key |
| Category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblhome:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | description | Constraints |
| H\_no | Int(255) | Home number | Primary key |
| H\_caption | Varchar(255) | caption | Not null |
| H\_img | Varchar(255) | image | Not null |
| H\_desc | Varchar(255) | description | Not null |

**tblpage**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | | Description | | Constraints | |
| ID | Int(255) | | ID | | Primary key | |
| PageType | Varchar(255) | | Type of page | | Not null | |
| PageTitle | Varchar(255) | | Title of page | | Not null | |
| PageDescription | Varchar(255) | | Description | | Not null | |
| Email | Varchar(255) | | Email | | Not null | |
| MobileNumber | Varchar(255) | | Mobile number | | Not null | |
| UpdationDate | Varchar(255) | Date of update | | | Not null |
|  | |
|  | |

**tblPhotoCategories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| cp\_no | Int(255) | Photo categories number | Primary key |
| category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblPhotoGallery**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| gc\_no | Int(255) | Photo gallery number | Primary key |
| category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblTeam**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| t\_id | Int(255) | team id | Primary key |
| t\_name | Varchar(30) | Team name | Not null |
| t\_work | Varchar(30) | Team work | Not null |
| t\_img | Varchar(30) | Team image | Not null |
| t\_intro | Int(255) | Team introduction | Not null |

**tbluserlogin**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | | Description | | Constraints |
| U\_f\_name | Varchar(30) | | User first name | | Not null |
| U\_s\_name | Varchar(30) | | User second name | | Not null |
| U\_mno | Int(255) | | User mobile No | | Not null |
| U\_email | Varchar(30) | | User email | | Not null |
| U\_id | Int(255) | | User id | | Not null |
| U\_pass | Varchar(30) | password | | Not null | |

**tblvideocategories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| cv\_no | Int(255) | Video category number | Primary key |
| category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblvideogallery**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| gv\_no | Int(255) | Video gallery number | Primary key |
| category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**tblvrcategories**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Constraints |
| cv\_no | Int(255) | Video number | Primary key |
| category | Varchar(30) | Category type | Not null |
| img | Varchar(30) | image | Not null |

**5.4 E-R diagram meaning**

An Entity relationship diagram s specialized graphic that illustrates the relationships between entities in a database.ER diagrams often use symbols to represent three different steps of information.Boxes are commonly used to represent entities.Diamonds are normally used to represent relationships and ovals are used to represent attributes.

**5.5 E-R Notation**

Entity

Relationship

Attribute

//

Weak Entity

Weak Entity relationship

Key Attribute

Composite Attribute

Multivalued Attribute

1. **Entity**

An **Entity** can be object,place,person or class.In E-R Diagram,an entity is represented using rectangles.Consider an example of an Organization.Employee,Manger,Department and many more can be taken as entities from an Organization.

**Weak entity**

Weak Entity is an entity that depends on another entity.Weak entity doesn’t have key attribute of their own.Double rectangle represents weak Entity.

1. **Attribute**

An attribute describes a property or characteristics of an entity.for example,name,age,address etc can be attributes of a student.An attribute is represented using eclipse.

**Key Attribute**

Key Attribute represents the main characteristics of an entity.it is used to represent primary key.Ellipse with underlying lines represents key Attribute.

**Composite Attribute**

An attribute can also have their own attributes.these attributes are known as Composite attribute.

1. **Relationship**

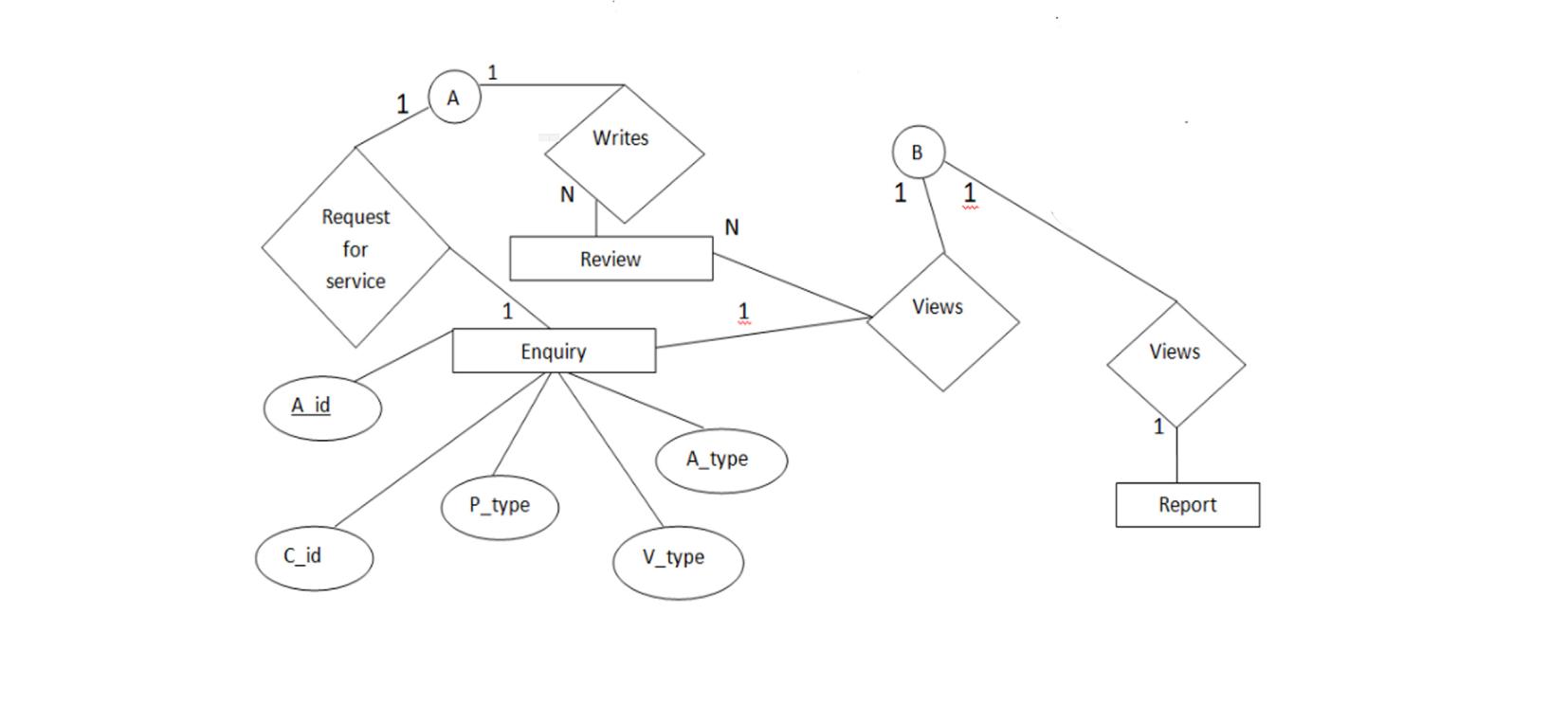
A relationship describes relations between entities.Relationship is represented using diamonds.

There are three types of relationship that exist between entities.

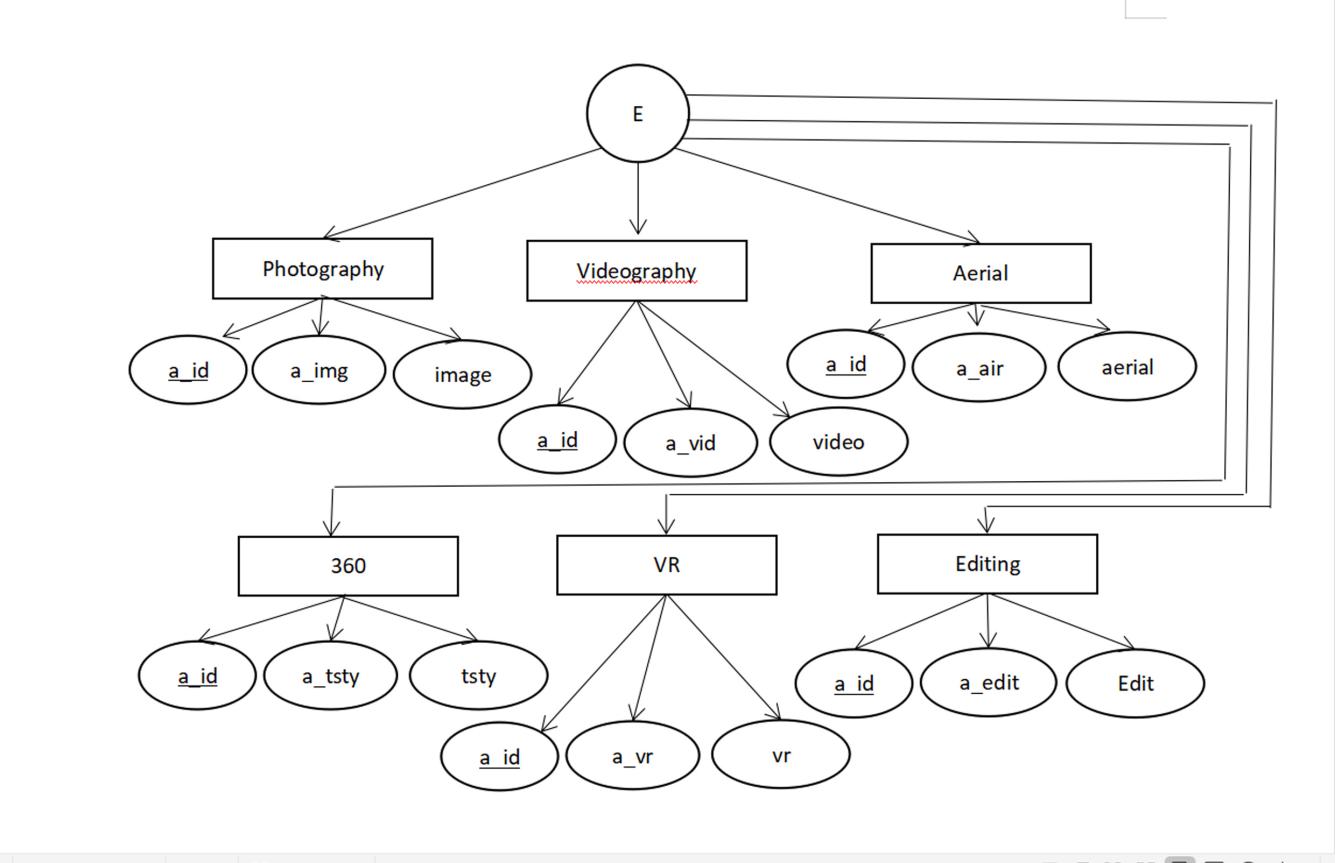
* Binary Relationship.
* Recursive Relationship.
* Ternary Relationship.
* Binary relationship.
* Binary relationship means relation between two Entities is further divided into three types.
* One to one: this type of relationship is rarely seen in real world.
* One to many:it reflects business rule that one entity is associated with many number of same entity.the example for this relation might sound a little weird,but one course will have one students.
* Many too many:it reflects business rule that many entities can be associated with many entity
* Recursive Relationship.
* When an entity is related with itself it is known as Recursive Relationship.
* Relationship of degree three is called ternary relationship.

**1.Entity-Relationship Diagram**









**DETAILED DESIGN**

**Chapter-6**

**DETAILED DESIGN**

**6.1 Introduction:**

The detail we decided the internal logic for the module,which implement the given specification detailed design is one of the design process for software items.Details process for software items.Detailed design,which is also known also as logic design,involves the internal sign of the module and how the specification of the module can be satisfied detailed design Is an extension of system design.

The first step before the detailed design or code for a module can be developed is that specification of the module be given precisely.once the module is precisely specified.the external logic for the module that will implement the given identification can be decided.

This Basic logic in detailed design is to specify the logic for the different modules that been specified during system design.specifying the logic will require developing an algorithm.

That will be implement the given specification.

This phase includes the specification of table structures,which makes us understand the internal logic of the system.the tables are described with their field names,data type,table constraints,description of each field etc.

**6.2 Structured English**

**Admin Login form:**

Begin

IF (Click On Login) THEN

Check The Admin Id And Password

IF (Admin Record Exists) THEN

Redirect Into Admin Page

Else

Display The Message”invalid Login Details Entered”

End If

End If

End

**Customer Registration Form:**

Begin

IF (Click On Registration) THEN

Customer Detail Form Will Be Displayed And User Is Allowed To Enter The Details

End If

IF (Click On Submit) THEN

Check For The Null Values And Other Entries If All Are Valid Then Save It To The Database

Else

Display Message “Invalid Entry”

END IF

End

**Logout Form:**

Begin

If(user Is Logged In)then

If(user Clicks On Logout Option”) Then

Destroy The Session Created For The Current User;

Lead User Panel;

End If

End If

End

**Service Booking module:**

Begin

IF (User Is Logged In As “Customer”) THEN

Display All Services;

IF (User Clicks On Any service) THEN

Display Service Information Through Service Table;

If(user Clicks On “Book Now”) THEN

Load Booking Module;

End If

End If

End If

End

**View admin profile**

Begin

If(click on View profile)then

View admin profile details will be displayed

Endif

End

**Edit Admin Profile**

Begin

Check Admin profile information exists in the profile database.

If(Admin information doesn’t exists)Then

Check for null values, internet connection and other valid fields and save the information into profile database and redirect to Homepage.

Else

Display message invalid fields.

Endif

End

**Add /edit Service**

Begin

If (click on add Service) Then

Check for null values.

If (All mandatory fields are filled correctly) Then

Save service information to the database.

Else

Display message ‘fields shouldn’t be empty’

End if

End if

If(click on Manage service)Then.

If(Same service exists)

Delete/edit service information in the database.

Else

Display the message “invalid service”.

End if

Endif

End

**Add/Manage Event**

Begin

If(click on add event) then

Check for null values

If(all Mandatory fields are filled correctly)Then

Save it to database.

Else

Display the message “fields shouldn’t be empty”

End if

If(click on manage event)then

Show all the events

If(click on delete)then

Delete event from database

End if

End if

End

**Add/Manage Gallery**

Begin

If(click on add Gallery) then

Check for null values

If(all Mandatory fields are filled correctly)Then

Save it to database.

Else

Display the message “fields shouldn’t be empty”

End if

If(click on manage Gallery)then

Show all gallery

If(click on delete)then

Delete gallery from database

End if

End if

End

**Service Booking module:**

Begin

IF (User Is Logged In As “Customer”) THEN

Display All Services;

IF (User Clicks On Any service) THEN

Display Service Information Through Service Table;

If(user Clicks On “Book Now”) THEN

If( Check if all mandatory fields are filled)

Add the information to database

Else

Display Please fill the form

End If

End If

End If

End If

End

**Admin Bookings search:**

Begin

IF (Click On Search bookings) THEN

Check the admin search record and check in database.

IF ( Record Exists) THEN

Redirect Into bookings Page

Else

IF (Bookings record doesn’t Exists) THEN

Display The Message”Entered Bookings doesn’t exists”

End If

End If

End If

End

**Queries**

Begin

If (click on queries)Then

Redirect to Q/A activity.

End if

If(click on to accept/deny)

Message to be sent through mail and the action should be stored in admin database.

End if

End

**User Login Form:**

Begin

IF (Click On Login) THEN

Check The Login Id And Password, If The User Exists Then Redirect Into Main Form

ELSE IF (User Does Not Exists) THEN

Display Message “Invalid Login ID” Or “Invalid Password”

End If

End

**Customer Registration Form:**

Begin

IF (Click On Registration) THEN

Customer Detail Form Will Be Displayed And User Is Allowed To Enter The Details

IF (Click On Submit) THEN

Check For The Null Values And Other Entries If All Are Valid Then Save It To The Database

Else

Display Message “Invalid Entry”

END IF

END IF

End

**User Login Form:**

Begin

IF (Click On Login) THEN

Check The Login Id And Password, If The User Exists Then Redirect Into Main Form

ELSE IF (User Does Not Exists) THEN

Display Message “Invalid Login ID” Or “Invalid Password”

End If

End

**Edit User Profile**

Begin

In homepage check users profile information exists in the profile database.

If(User information doesn’t exists)Then

Check for null values and other valid fields and save the information into profile database and redirect to Homepage.

Else

Display message invalid fields.

Endif

End

**View user profile**

Begin

If(click on View profile)then

View user profile details will be displayed

Endif

End

**View Gallery**

Begin

If(click on Gallery)then

Admins gallery details will be displayed.

Endif

End

**Forgot password**

Begin

If(click on forgot password) then

Ask email

If(email is in database)

Send password to mail

Else

Display email is invalid

End if

Endif

End

**CHAPTER-7**

**SYSTEM CODING**

* 1. **Introduction:**

The main goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language.it is then executed on the computer to verify whether the design is correct or not.in coding phase the output is code can be extremely useful in enhancing the understandability .internal documentation of code is done using comments in the program.comments are textual statements that ate meant for the programming reader and are not executed.

The coding phase effects both testing and maintenance phases.well write code can reduce the testing and maintenance effort.the goal of coding should be reduce the testing and maintenance effort.

* 1. **Module Coding**

Home Page

<?php

session\_start();

error\_reporting(0);

 include('../include/dbconnection.php');

 $query=mysqli\_query($con,"select \* from tbluserlogin");

?>

<!DOCTYPE html>

<html>

<head>

    <title>Capture It</title>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>

        /\* slider code \*/

        /\* Login code\*/

        body {

            font-family: Arial, Helvetica, sans-serif;

        }

        /\* Full-width input fields \*/

        input[type=text],

        input[type=password] {

            width: 100%;

            padding: 12px 20px;

            margin: 8px 0;

            display: inline-block;

            border: 1px solid #ccc;

            box-sizing: border-box;

        }

        /\* Set a style for all buttons \*/

        button {

            background-color: #4CAF50;

            color: white;

            padding: 14px 20px;

            margin: 8px 0;

            border: none;

            cursor: pointer;

            width: 100%;

        }

        button:hover {

            opacity: 0.8;

        }

        /\* Extra styles for the cancel button \*/

        .cancelbtn {

            width: auto;

            padding: 10px 18px;

            background-color: #f44336;

        }

        /\* Center the image and position the close button \*/

        .imgcontainer {

            text-align: center;

            margin: 24px 0 12px 0;

            position: relative;

        }

        img.avatar {

            width: 40%;

            border-radius: 50%;

        }

        .container {

            padding: 16px;

        }

        span.psw {

            float: right;

            padding-top: 16px;

        }

        /\* The Modal (background) \*/

        .modal {

            display: none;

            /\* Hidden by default \*/

            position: fixed;

            /\* Stay in place \*/

            z-index: 1;

            /\* Sit on top \*/

            left: 0;

            top: 0;

            width: 100%;

            /\* Full width \*/

            height: 100%;

            /\* Full height \*/

            overflow: auto;

            /\* Enable scroll if needed \*/

            background-color: rgb(0, 0, 0);

            /\* Fallback color \*/

            background-color: rgba(0, 0, 0, 0.4);

            /\* Black w/ opacity \*/

            padding-top: 60px;

        }

        /\* Modal Content/Box \*/

        .modal-content {

            background-color: #fefefe;

            margin: 5% auto 15% auto;

            /\* 5% from the top, 15% from the bottom and centered \*/

            border: 1px solid #888;

            width: 80%;

            /\* Could be more or less, depending on screen size \*/

        }

        /\* The Close Button (x) \*/

        .close {

            position: absolute;

            right: 25px;

            top: 0;

            color: #000;

            font-size: 35px;

            font-weight: bold;

        }

        .close:hover,

        .close:focus {

            color: red;

            cursor: pointer;

        }

        /\* Add Zoom Animation \*/

        .animate {

            -webkit-animation: animatezoom 0.6s;

            animation: animatezoom 0.6s

        }

        @-webkit-keyframes animatezoom {

            from {

                -webkit-transform: scale(0)

            }

            to {

                -webkit-transform: scale(1)

            }

        }

        @keyframes animatezoom {

            from {

                transform: scale(0)

            }

            to {

                transform: scale(1)

            }

        }

        /\* Change styles for span and cancel button on extra small screens \*/

        @media screen and (max-width: 300px) {

            span.psw {

                display: block;

                float: none;

            }

            .cancelbtn {

                width: 100%;

            }

        }

        /\* slider code \*/

        \* {

            box-sizing: border-box;

        }

        body {

            font-family: Verdana, sans-serif;

        }

        .mySlides {

            display: none;

        }

        img {

            vertical-align: middle;

        }

        /\* Slideshow container \*/

        .slideshow-container {

            max-width: 1000px;

            position: relative;

            margin: auto;

        }

        /\* Caption text \*/

        .text {

            color: #f2f2f2;

            font-size: 15px;

            padding: 8px 12px;

            position: absolute;

            bottom: 8px;

            width: 100%;

            text-align: center; }

        /\* Number text (1/3 etc) \*/

        .numbertext {

            color: #f2f2f2;

            font-size: 12px;

            padding: 8px 12px;

            position: absolute;

            top: 0;

        }

        /\* The dots/bullets/indicators \*/

        .dot {

            height: 15px;

            width: 15px;

            margin: 0 2px;

            background-color: #bbb;

            border-radius: 50%;

            display: inline-block;

            transition: background-color 0.6s ease;

        }

        .active {

            background-color: #717171;

        }

        /\* Fading animation \*/

        .fade {

            -webkit-animation-name: fade;

            -webkit-animation-duration: 2s;

            animation-name: fade;

            animation-duration: 2s;

        }

        @-webkit-keyframes fade {

            from {

                opacity: .4

            }

            to {

                opacity: 1

            }

        }

        @keyframes fade {

            from {

                opacity: .4

            }

            to {

                opacity: 1

            }

        }

        /\* On smaller screens, decrease text size \*/

        @media only screen and (max-width: 300px) {

            .text {

                font-size: 11px

            }

        }

        img {

            display: block;

            margin-left: auto;

            margin-right: auto;

        }

    </style>

    <link href="css/bootstrap.css" rel='stylesheet' type='text/css' />

    <!-- jQuery (Bootstrap's JavaScript plugins) -->

    <script src="js/jquery.min.js"></script>

    <script type="application/x-javascript">

        addEventListener("load", function() {

            setTimeout(hideURLbar, 0);

        }, false);

        function hideURLbar() {

            window.scrollTo(0, 1);

        }

    </script>

    <!-- Custom Theme files -->

    <link href="css/style.css" rel="stylesheet" type="text/css" media="all" />

    <link href='http://fonts.googleapis.com/css?family=Monoton' rel='stylesheet' type='text/css'>

    <link href='http://fonts.googleapis.com/css?family=Raleway' rel='stylesheet' type='text/css'>

</head>

<body>

    <!---->

    <?php include\_once('includes/header.php');?>

    <?php include\_once('includes/slider.php');?>

    <!---->

    <script src="js/responsiveslides.min.js"></script>

    <script>

        // You can also use "$(window).load(function() {"

        $(function() {

            $("#slider").responsiveSlides({

                auto: true,

                manualControls: '#slider3-pager',

            });

        });

 </script>

 <?php include\_once('includes/footer.php');?>  <!---->

</body></html>

**User login form**

<?php

 session\_start();

 ?>

<!DOCTYPE html>

<html>

<head>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>

        @import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");

        \* {

            box-sizing: border-box;

        }

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            margin: 0;

            min-height: 100vh;

            font-family: 'Montserrat', sans-serif;

            background: rgb(2, 0, 36);

            background: linear-gradient(90deg,

                    rgba(2, 0, 36, 1) 0%,

                    rgba(6, 215, 152, 1) 0%,

                    rgba(135, 219, 213, 1) 30%,

                    rgba(32, 204, 158, 1) 70%);

        }

        .container {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            padding: 50px 100px;

            background-color: #fff;

            border-radius: 5px;

            width: 25em;

            height: 40em;

        }

        .items {

            display: flex;

            flex-direction: column;

        }

        .items input {

            margin-bottom: 30px;

            padding: 2px 0px;

            padding-bottom: 5px;

            width: 250px;

            background-color: transparent;

            border: none;

            outline: none;

            border-bottom: 1.5px solid;

            border-bottom-color: grey;

            font-weight: 700;

            color: #585858;

            opacity: 0.55;

        }

        #title {

            margin-bottom: 50px;

            letter-spacing: 5px;

            text-align: center;

            opacity: 0.5;

        }

        #button {

            padding: 20px 30px;

            margin-top: 30px;

            outline: none;

            border: none;

            font-size: medium;

            font-weight: 600;

            letter-spacing: 5px;

            border-radius: 5px;

            cursor: pointer;

            background: linear-gradient(90deg,

                    rgba(2, 0, 36, 1) 0%,

                    rgba(6, 215, 152, 1) 0%,

                    rgba(32, 204, 158, 1) 30%,

                    rgba(135, 219, 213, 1) 70%);

            color: grey;

        }

        #button:active {

            transform: translateY(2px);

        }

        #button text {

            opacity: 0.5;

        }

        #sign {

            margin-top: 4em;

            text-align: center;

            letter-spacing: 5px;

            cursor: pointer;

            color: rgb(11, 194, 142);

            font-weight: bolder;

            text-decoration: none;

        }

        #sign:active {

            transform: translateY(2px);

        }

        ::placeholder {

            color: grey;

            text-align: center;

            letter-spacing: 2px;

        }

    </style>

</head>

<body>

    <div class="container">

        <h2 id="title">Login</h2>

 <form action="u\_login.php" method="post">

        <div class="items">

            <input type="text" id="form-text" placeholder="Username" name="email" />

            <input type="password" id="form-password" placeholder="Password" name="pass" />

            <button id="button"  type="submit" value="submit" name="submit">Send</button>

            <a id="sign" type="submit" href="sign\_up.php">Sign Up</a>

            <a id="sign" type="submit" href="forgot\_password.php">Fogot password?</a>

        </div>

    </form>

</div>

</body>

</html>

<?php

include('includes/dbconnection.php');

if(isset($\_POST['submit'])) {

  $email=$\_POST['email'];

  $pass=$\_POST['pass'];

    $sql = "SELECT \* FROM tbluserlogin where u\_email='$email' && u\_pass='$pass'";

    $query = $dbh -> prepare($sql);

    $query -> execute();

    $results = $query -> fetchAll(PDO::FETCH\_OBJ);

if($query -> rowCount() > 0){

   echo $\_SESSION['email']=$email;

   header('location:index.php');

    }

else{

   echo "<script>alert('Email or password is incorrect, Please try again')</script>";

}

}

?>

**Sign up form**

<?php

session\_start();

?>

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

@import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");

\* {

box-sizing: border-box;

}

body {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

margin: 0;

min-height: 100vh;

font-family: 'Montserrat', sans-serif;

background: rgb(2, 0, 36);

background: linear-gradient(90deg,

rgba(2, 0, 36, 1) 0%,

rgba(6, 215, 152, 1) 0%,

rgba(135, 219, 213, 1) 30%,

rgba(32, 204, 158, 1) 70%);

}

.container {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding: 50px 100px;

background-color: #fff;

border-radius: 5px;

width: 25em;

height: 40em;

}

.items {

display: flex;

flex-direction: column;

}

.items input {

margin-bottom: 30px;

padding: 2px 0px;

padding-bottom: 5px;

width: 250px;

background-color: transparent;

border: none;

outline: none;

border-bottom: 1.5px solid;

border-bottom-color: grey;

font-weight: 700;

color: #585858;

opacity: 0.55;

}

#title {

margin-bottom: 50px;

letter-spacing: 5px;

text-align: center;

opacity: 0.5;

}

#button {

padding: 20px 30px;

margin-top: 30px;

outline: none;

border: none;

font-size: medium;

font-weight: 600;

letter-spacing: 5px;

border-radius: 5px;

cursor: pointer;

background: linear-gradient(90deg,

rgba(2, 0, 36, 1) 0%,

rgba(6, 215, 152, 1) 0%,

rgba(32, 204, 158, 1) 30%,

rgba(135, 219, 213, 1) 70%);

color: grey;

}

#button:active {

transform: translateY(2px);

}

#button text {

opacity: 0.5;

}

#sign {

margin-top: 4em;

text-align: center;

letter-spacing: 5px;

cursor: pointer;

color: rgb(11, 194, 142);

font-weight: bolder;

text-decoration: none;

}

#sign:active {

transform: translateY(2px);

}

::placeholder {

color: grey;

text-align: center;

letter-spacing: 2px;

}

</style>

</head>

<body>

<div class="container">

<h2 id="title">Create Acoount</h2>

<form action="sign\_up.php" method="post">

<div class="items">

<input type="text" id="form-text" placeholder="First Name" name="first\_name" />

<input type="text" id="form-text" placeholder="Second Name" name="last\_name" />

<input type="text" id="form-text" placeholder="Email" name="email" />

<input type="text" id="form-text" placeholder="Phone no." name="mobile\_number" /><input type="password" id="form-password" placeholder="Password" name="password" />

<button id="button" type="submit" value="submit" name="submit">Create account</button>

<a id="sign" type="submit" href="u\_login.php">Login</a>

</div>

</form>

</div>

</body>

</html>

<?php

include('includes/dbconnection.php');

if(isset($\_POST['submit'])){

$first\_name=$\_POST['first\_name'];

$last\_name=$\_POST['last\_name'];

$email=$\_POST['email'];

$mobile\_number=$\_POST['mobile\_number'];

$password=$\_POST['password'];

$sql = "INSERT INTO `tbluserlogin`(`u\_email`,`u\_pass`,`u\_f\_name`,`u\_s\_name`,`u\_mno`)

VALUES(:email,:password,:first\_name,:last\_name,:mobile\_number)";

$query = $dbh -> prepare($sql);

$query->bindParam(':first\_name',$first\_name,PDO::PARAM\_STR);

$query->bindParam(':last\_name',$last\_name,PDO::PARAM\_STR);

$query->bindParam(':email',$email,PDO::PARAM\_STR);

$query->bindParam(':password',$password,PDO::PARAM\_INT);

$query->bindParam(':mobile\_number',$mobile\_number,PDO::PARAM\_INT);

$query -> execute();

if($query) {

echo "<script>alert('Now you can login')</script>";

header('location:u\_login.php');

}else{

echo '<script language="javascript">';

echo 'alert("Sorry!! Try Again")';

echo '</script>';

}

}

?>

**Logout form**

<?php

**session\_start();**

**session\_unset();**

**header('location:index.php');**

?>

**Forgot password form**

<?php

 session\_start();

 ?>

<!DOCTYPE html>

<html>

<head>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>

        @import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");

        \* {

            box-sizing: border-box;

        }

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            margin: 0;

            min-height: 100vh;

            font-family: 'Montserrat', sans-serif;

            background: rgb(2, 0, 36);

            background: linear-gradient(90deg,

                    rgba(2, 0, 36, 1) 0%,

                    rgba(6, 215, 152, 1) 0%,

                    rgba(135, 219, 213, 1) 30%,

                    rgba(32, 204, 158, 1) 70%);

        }

        .container {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            padding: 50px 100px;

            background-color: #fff;

            border-radius: 5px;

            width: 25em;

            height: 40em;

        }

        .items {

            display: flex;

            flex-direction: column;

        }

        .items input {

            margin-bottom: 30px;

            padding: 2px 0px;

            padding-bottom: 5px;

            width: 250px;

            background-color: transparent;

            border: none;

            outline: none;

            border-bottom: 1.5px solid;

            border-bottom-color: grey;

            font-weight: 700;

            color: #585858;

            opacity: 0.55;

        }

        #title {

            margin-bottom: 50px;

            letter-spacing: 5px;

            text-align: center;

            opacity: 0.5;

        }

        #button {

            padding: 20px 30px;

            margin-top: 30px;

            outline: none;

            border: none;

            font-size: medium;

            font-weight: 600;

            letter-spacing: 5px;

            border-radius: 5px;

            cursor: pointer;

            background: linear-gradient(90deg,

                    rgba(2, 0, 36, 1) 0%,

                    rgba(6, 215, 152, 1) 0%,

                    rgba(32, 204, 158, 1) 30%,

                    rgba(135, 219, 213, 1) 70%);

            color: grey;

        }

        #button:active {

            transform: translateY(2px);

        }

        #button text {

            opacity: 0.5;

        }

        #sign {

            margin-top: 4em;

            text-align: center;

            letter-spacing: 5px;

            cursor: pointer;

            color: rgb(11, 194, 142);

            font-weight: bolder;

            text-decoration: none;

        }

        #sign:active {

            transform: translateY(2px);

        }

        ::placeholder {

            color: grey;

            text-align: center;

            letter-spacing: 2px;

        }

    </style>

</head>

<body>

    <div class="container">

        <h2 id="title">Forgot Password</h2>

        <form action="forgot\_password.php" method="post">

            <div class="items">

                <input type="text" id="form-text" placeholder="Enter your email" name="email" />

                <button id="button" type="submit" value="submit" name="submit">Reset Password</button>

                <a id="sign" type="submit" href="u\_login.php">Login</a>

                <a id="sign" type="submit" href="sign\_up.php">Sign Up</a>

            </div>

        </form>

    </div>

</body>

</html>

<?php

// Import PHPMailer classes into the global namespace

// These must be at the top of your script, not inside a function

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

// Load Composer's autoloader

require 'vendor/autoload.php';

include('includes/dbconnection.php');

if(isset($\_POST['submit'])) {

  $email=$\_POST['email'];

    $sql = "SELECT \* FROM tbluserlogin where u\_email='$email'";

    $query = $dbh -> prepare($sql);

    $query -> execute();

    $results = $query -> fetchAll(PDO::FETCH\_OBJ);

if($query -> rowCount() > 0){

    $sql = "SELECT \* FROM tbluserlogin where u\_email='$email'";

    $query = $dbh -> prepare($sql);

    $query -> execute();

    $results = $query -> fetchColumn(5);

// Instantiation and passing `true` enables exceptions

$mail = new PHPMailer(true);

try {

    //Server settings

    //$mail->SMTPDebug = 2;                      // Enable verbose debug output

    $mail->isSMTP();                                            // Send using SMTP

    $mail->Host       = 'smtp.gmail.com';                    // Set the SMTP server to send through

    $mail->SMTPAuth   = true;                                   // Enable SMTP authentication

    $mail->Username   = 'captureitsmc@gmail.com';                     // SMTP username

    $mail->Password   = '9731424843';                               // SMTP password

    $mail->SMTPSecure = 'tls';         // Enable TLS encryption; `PHPMailer::ENCRYPTION\_SMTPS` also accepted

    $mail->Port       = 587;                                    // TCP port to connect to

    //Recipients

    $mail->setFrom('captureitsmc@gmail.com', 'test');

    $mail->addAddress($email);     // Add a recipient

// Name is optional

    // Content

    $mail->isHTML(true);                                  // Set email format to HTML

    $mail->Subject = 'TEST';

    $mail->Body    = $results;

    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

    $mail->send();

    echo 'Message has been sent';

} catch (Exception $e) {

    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";

}

}

else{

   echo "<script>alert('Email or password is incorrect, Please try again')</script>";

}

}

?>

**Booking services**

<?php

session\_start();

error\_reporting(0);

include('includes/dbconnection.php');

    if(isset($\_POST['submit']))

  {

    $bid=$\_GET['bookid'];

 $name=$\_POST['name'];

  $mobnum=$\_POST['mobnum'];

 $email=$\_POST['email'];

 $edate=$\_POST['edate'];

 $est=$\_POST['est'];

  $eetime=$\_POST['eetime'];

 $vaddress=$\_POST['vaddress'];

 $eventtype=$\_POST['eventtype'];

 $addinfo=$\_POST['addinfo'];

 $bookingid=mt\_rand(100000000, 999999999);

$sql="insert into tblbooking(BookingID,ServiceID,Name,MobileNumber,Email,EventDate,EventStartingtime,EventEndingtime,VenueAddress,EventType,AdditionalInformation)values(:bookingid,:bid,:name,:mobnum,:email,:edate,:est,:eetime,:vaddress,:eventtype,:addinfo)";

$query=$dbh->prepare($sql);

$query->bindParam(':bookingid',$bookingid,PDO::PARAM\_STR);

$query->bindParam(':bid',$bid,PDO::PARAM\_STR);

$query->bindParam(':name',$name,PDO::PARAM\_STR);

$query->bindParam(':mobnum',$mobnum,PDO::PARAM\_STR);

$query->bindParam(':email',$email,PDO::PARAM\_STR);

$query->bindParam(':edate',$edate,PDO::PARAM\_STR);

$query->bindParam(':est',$est,PDO::PARAM\_STR);

$query->bindParam(':eetime',$eetime,PDO::PARAM\_STR);

$query->bindParam(':vaddress',$vaddress,PDO::PARAM\_STR);

$query->bindParam(':eventtype',$eventtype,PDO::PARAM\_STR);

$query->bindParam(':addinfo',$addinfo,PDO::PARAM\_STR);

 $query->execute();

   $LastInsertId=$dbh->lastInsertId();

   if ($LastInsertId>0) {

    echo '<script>alert("Your Booking Request Has Been Send. We Will Contact You Soon")</script>';

echo "<script>window.location.href ='services.php'</script>";

  }

  else

    {

         echo '<script>alert("Something Went Wrong. Please try again")</script>';

    }

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Capture it || book</title>

<link href="css/bootstrap.css" rel='stylesheet' type='text/css' />

<!-- Custom Theme files -->

<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />

<link rel="stylesheet" href="css/touchTouch.css" type="text/css" media="all" />

<!-- Custom Theme files -->

<script src="js/jquery.min.js"></script>

<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>

<!--webfont-->

<link href='http://fonts.googleapis.com/css?family=Monoton' rel='stylesheet' type='text/css'>

<link href='http://fonts.googleapis.com/css?family=Raleway' rel='stylesheet' type='text/css'>

<!---//End-css-style-switecher----->

<script type="text/javascript" src="js/jquery.fancybox.js"></script>

        <link rel="stylesheet" type="text/css" href="css/jquery.fancybox.css" media="screen" />

       <script type="text/javascript">

            $(document).ready(function() {

                /\*

                 \*  Simple image gallery. Uses default settings

                 \*/

                $('.fancybox').fancybox();

            });

        </script>

</head>

<body>

<?php include\_once('includes/header.php');?>

<div class="contact content">

     <div class="container">

         <ol class="breadcrumb">

          <li><a href="index.php">Home</a></li>

          <li class="active">Book Services</li>

         </ol>

             <div class="contact-grids">

                 <div class="col-md-6 contact-left">

                     <p>Book Your Events now. </p>

                     <form method="post">

                         <ul>

                             <li class="text-info" style="color:black">Name: </li>

                             <li><input type="text" class="form-control" name="name" required="true"></li>

                         </ul>

                         <ul>

                             <li class="text-info" style="color:black">Email: </li>

                             <li><input type="email" class="form-control" name="email" required="true"></li>

                         </ul>

                         <ul>

                             <ul>

                             <li class="text-info" style="color:black">Mobile Number: </li>

                             <li><input type="text" class="text" name="mobnum" required="true" maxlength="10" pattern="[0-9]+"></li>

                         </ul>

                             <li class="text-info" style="color:black">Event Date: </li>

                             <li><input type="date" class="form-control" name="edate" required="true"></li>

                         </ul>

                         <ul>

                             <li class="text-info" style="color:black">Event Starting Time:</li>

                             <li><select type="text" class="form-control" name="est" required="true">

                                <option value="">Select Starting Time</option>

                                <option value="1 a.m">1 a.m</option>

                                <option value="2 a.m">2 a.m</option>

                                <option value="3 a.m">3 a.m</option>

                                <option value="4 a.m">4 a.m</option>

                                <option value="5 a.m">5 a.m</option>

                                <option value="6 a.m">6 a.m</option>

                                <option value="7 a.m">7 a.m</option>

                                <option value="8 a.m">8 a.m"</option>

                                <option value="9 a.m">9 a.m</option>

                                <option value="10 a.m">10 a.m</option>

                                <option value="11 a.m">11 a.m</option>

                                <option value="12 p.m">12 a.m</option>

                                <option value="1 p.m">1 p.m</option>

                                <option value="2 p.m">2 p.m</option>

                                <option value="3 p.m">3 p.m</option>

                                <option value="4 p.m">4 p.m</option>

                                <option value="5 p.m">5 p.m</option>

                                <option value="6 p.m">6 p.m</option>

                                <option value="7 p.m">7 p.m</option>

                                <option value="8 p.m">8 p.m</option>

                                <option value="9 p.m">9 p.m</option>

                                <option value="10 p.m">10 p.m</option>

                                <option value="10 p.m">10 p.m</option>

                                <option value="12 a.m">12 a.m</option>

                             </select></li>

                         </ul>

                         <ul>

                             <li class="text-info"style="color:black">Event Finish Time:</li>

                             <li><select type="text" class="form-control" name="eetime" required="true">

                                <option value="">Select Finish Time</option>

                                <option value="1 a.m">1 a.m</option>

                                <option value="2 a.m">2 a.m</option>

                                <option value="3 a.m">3 a.m</option>

                                <option value="4 a.m">4 a.m</option>

                                <option value="5 a.m">5 a.m</option>

                                <option value="6 a.m">6 a.m</option>

                                <option value="7 a.m">7 a.m</option>

                                <option value="8 a.m">8 a.m"</option>

                                <option value="9 a.m">9 a.m</option>

                                <option value="10 a.m">10 a.m</option>

                                <option value="11 a.m">11 a.m</option>

                                <option value="12 p.m">12 a.m</option>

                                <option value="1 p.m">1 p.m</option>

                                <option value="2 p.m">2 p.m</option>

                                <option value="3 p.m">3 p.m</option>

                                <option value="4 p.m">4 p.m</option>

                                <option value="5 p.m">5 p.m</option>

                                <option value="6 p.m">6 p.m</option>

                                <option value="7 p.m">7 p.m</option>

                                <option value="8 p.m">8 p.m</option>

                                <option value="9 p.m">9 p.m</option>

                                <option value="10 p.m">10 p.m</option>

                                <option value="10 p.m">10 p.m</option>

                                <option value="12 a.m">12 a.m</option>

                             </select></li>

                         </ul>

                         <ul>

                             <li class="text-info" style="color:black">Venue Address:</li>

                             <li><textarea type="text" class="form-control" name="vaddress" required="true" ></textarea></li>

                         </ul>

                         <ul>

                             <li class="text-info" style="color:black">Type of Event:</li>

                             <li><select type="text" class="form-control" name="eventtype" required="true" >

                                <option value="">Choose Event Type</option>

                                <?php

$sql2 = "SELECT \* from   tbleventtype ";

$query2 = $dbh -> prepare($sql2);

$query2->execute();

$result2=$query2->fetchAll(PDO::FETCH\_OBJ);

foreach($result2 as $row)

{

    ?>

<option value="<?php echo htmlentities($row->EventType);?>"><?php echo htmlentities($row->EventType);?></option>

 <?php } ?>

                             </select></li>

                         </ul>

                         <ul>

                             <li class="text-info" style="color:black">Additional Information:</li>

                             <li><textarea type="text" class="form-control" name="addinfo" required="true"></textarea></li>

                         </ul>

                         <input type="submit" name="submit" value="Book">

                     </form>

                 </div>

                 <div class="col-md-6 contact-right">

                        <div class="contact-map">

                        <img src="images/abt.png" class="img-responsive" height="900" width="500" alt=""/>

                        </div>

                 </div>

                 <div class="clearfix"></div>

             </div>

         </div>

        <?php include\_once('includes/footer.php');?>

     </div>

</div>

<!---->

<!---->

</body>

</html>

**TESTING**

**Chapter-8**

**System testing**

**8.1 Introduction**

Testing is the process of detecting errors.Testing performs a very special role for quality testing and for assuring the reliability of software.The result of testing are used for later on during maintainance also.

**8.2 Psychology of testing**

The aim of testing is often to demonstrate the a program works by showing that it has no works.the basic of testing phase is to detect the errors that may be present in the program hence one should not start testing with the intent to show that a program does not work.testing is the process of executing a program with the intent of finding errors.

**8.3 Testing objectives**

The main objective of the testing is to cover a host of errors,systematically and with the minimum effort and time.starting formally,we can say,testing is a process of executing program with the intent of finding an error.

* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has high probability of finding an error,if it exists.
* The test is inadequate to detect possibilities present errors.
* The software more or less confirms to be quality and reliable standards.

**8.4 System testing**

Software testing is a critical element of software quality assurance and represents.

The ultimate review if a specification design,coding.the testing phase involves the testing of system using various test data,the system under study is tested.

Those test data error were found and connected by following testing steps and corrections are recorded for future references.thus a series testing is performed on the system before it is ready for implementation.

**8.5 Levels of testing**

* **Unit testing**
* **Integration testing**
* **Validation testing**
* **Output testing**
* **User acceptance testing**

**Unit testing:**

unit testing focuses verification efforts on the smallest unit of software .i.e the module .using detailed design and the process specification testing is done to uncover within a boundary of the module all the module must be successful in the unit test before start of integration testing begins.

In this project,unit testing is very much essential because the error is accurate.unit testing is the testing of different units or modules of a system in isolation.testing is done to check whether each module In the software works properly.

**Code walkthrough**

In this phase of testing the code is thorowly checked for the discrepancies that may occur in the code such as redundancy of the code,deviations in naming convention etc.

**Integration testing**

In this all the code modules are put together and tested for a desired output.the module unit tested is integration and tested.all the modules are combined in this resting step.the the entire program is tested as whole.the integration testing is done.

**Output testing**

Feeding sample valid inputs image and camparing the ratio obtained in this compressedoutput image with the expected ratio conduct it.correctness of the output depends on the inputted image.

**System testing**

In system testing entire system is tested as whole with all forms,code and class modules.after the integration testing whole of the system is tested in different environments and it is found that the system well without giving any runtime error.hence after the testing it is concluded that the system will work fine in all environments.

**Test cases**

**Admin login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial**  **NO.** | **Condition**  **To be tested** | **Test data** | **Expected**  **output** | **remarks** |
| 1. | If any field in the form is empty | Value of form | Alert the user to enter all the fields and then proceed. | Successful |
| 2. | If the Login id and password does not match | Login id, password | Alert the user that login id and password are not matching and stay in the same page | Successful |

**User login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial  NO. | Condition  To be tested | Test data | Expected  output | remarks |
| 1. | If any field in the form is empty | Value of form fields | Alert the user to enter all the fields and then proceed. | Successful |
| 2. | If the Login id and password does not match | Login id, password | Alert the user that login id and password are not matching and stay in the same page | Successful |

**User registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial  NO. | Condition  To be tested | Test data | Expected  output | remarks |
| 1. | If any field in the form is empty | Value of form | Alert the user to enter all the fields and then proceed. | Successful |
| 2. | If mobile contact contain more than 10 digits | Contact no | Alert the user to enter only 10 digits. | Successful |
| 3. | If the email id is invalid | Email id | Alert the user to enter a valid email id and return to the same page | Successful |
| 4. | If the confirm password and password does not match | Password | Alert the user to enter matching password and confirm password | Successful |

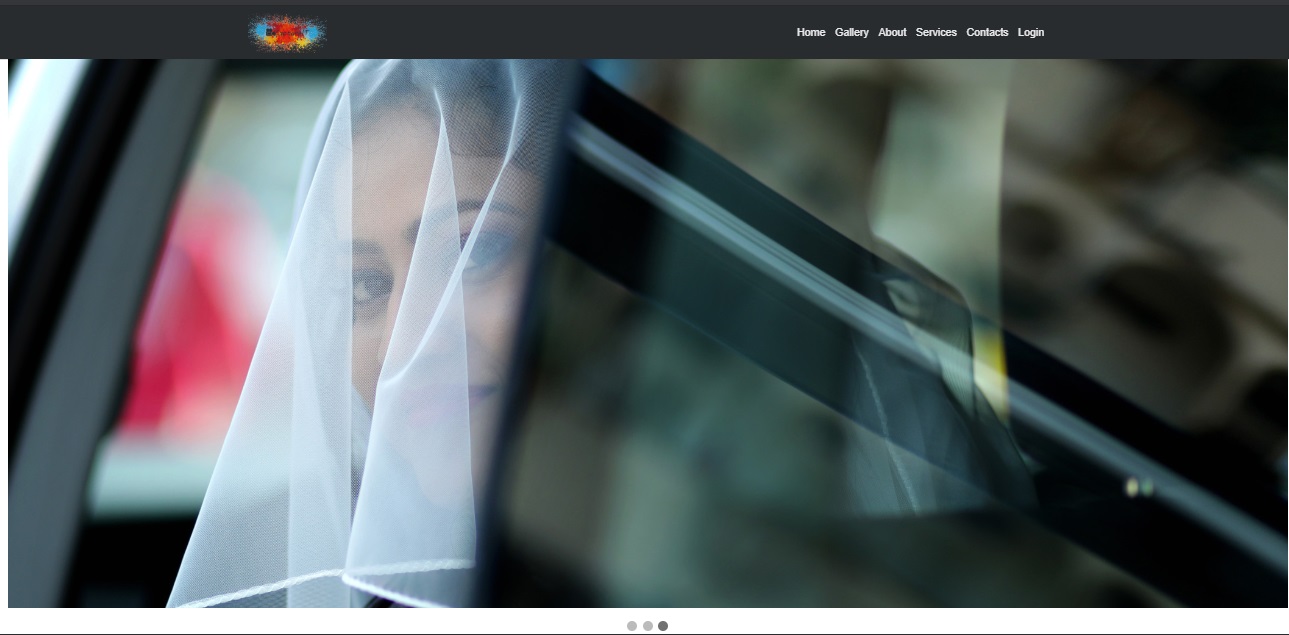
**Booking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial  NO. | Condition  To be tested | Test data | Expected  output | remarks |
| 1. | If any field in the form is empty | Value of form | Alert the user to enter all the fields and then proceed. | Successful |
| 2. | If mobile contact contain more than 10 digits | Contact no | Alert the user to enter only 10 digits. | Successful |
| 3. | If the email id is invalid | Email id | Alert the user to enter a valid email id and return to the same page | Successful |
| 4. | If Event date is invalid | Event date | Alert the user to enter valid date | Successful |

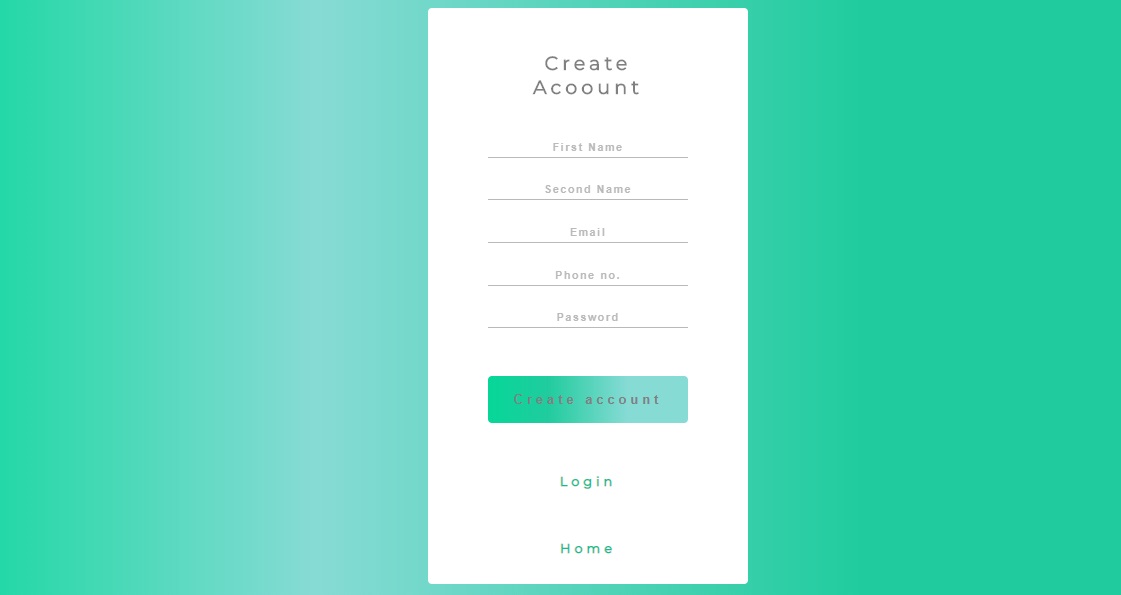
**Chapter 9**

**Snapshots**

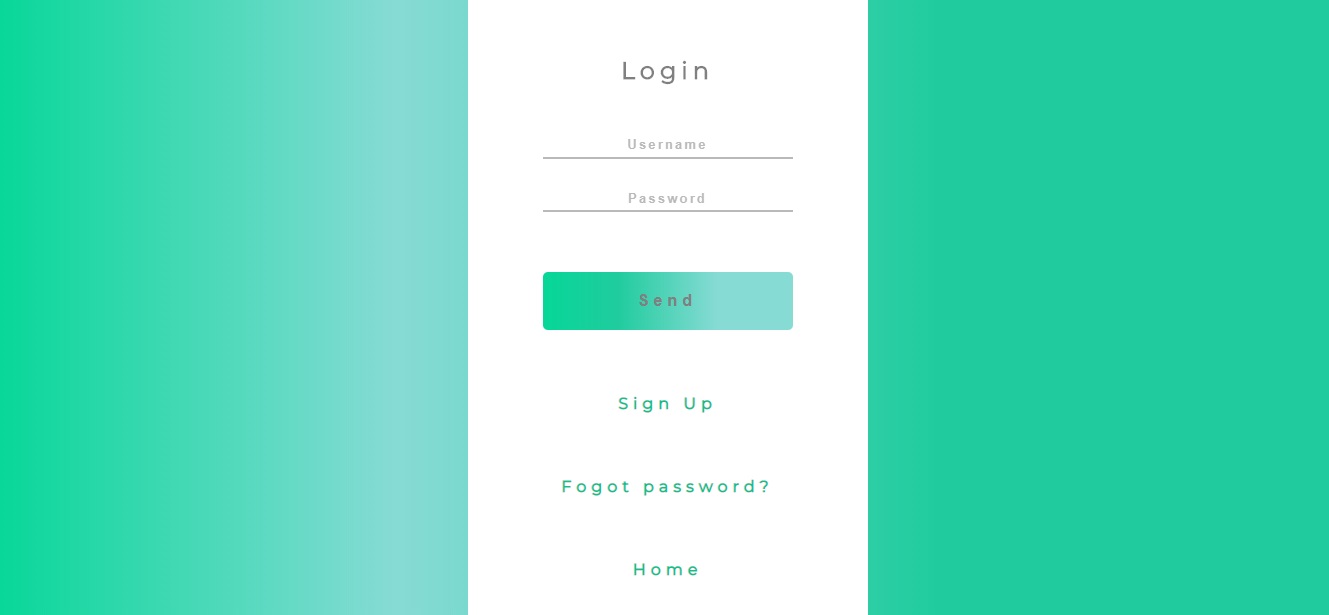
Home page



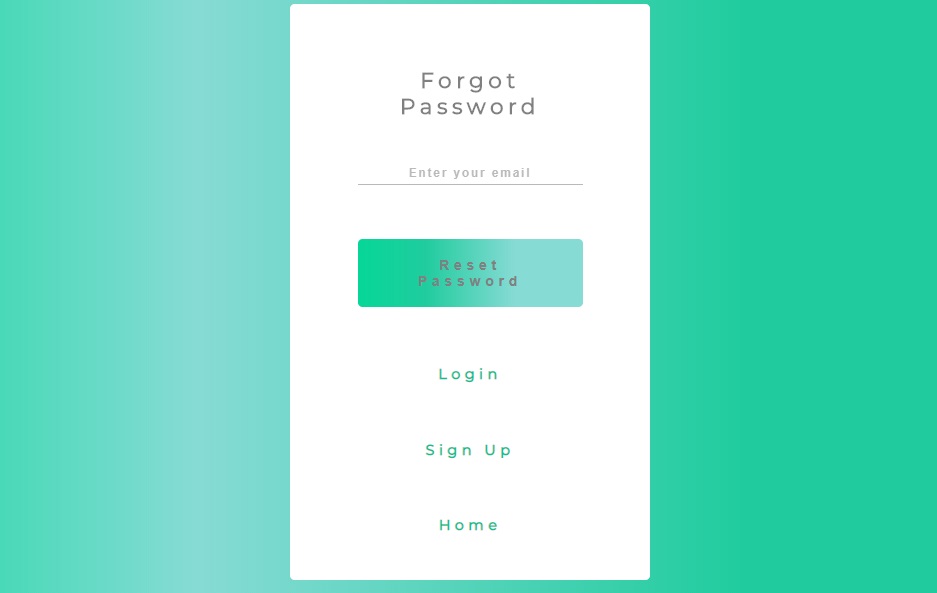
Signup page



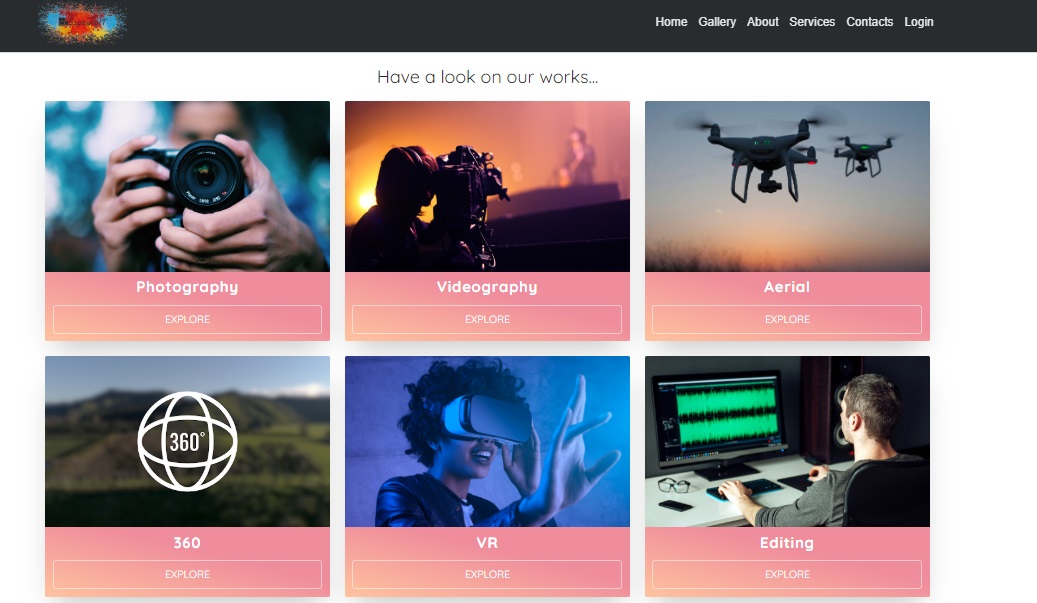
Login page



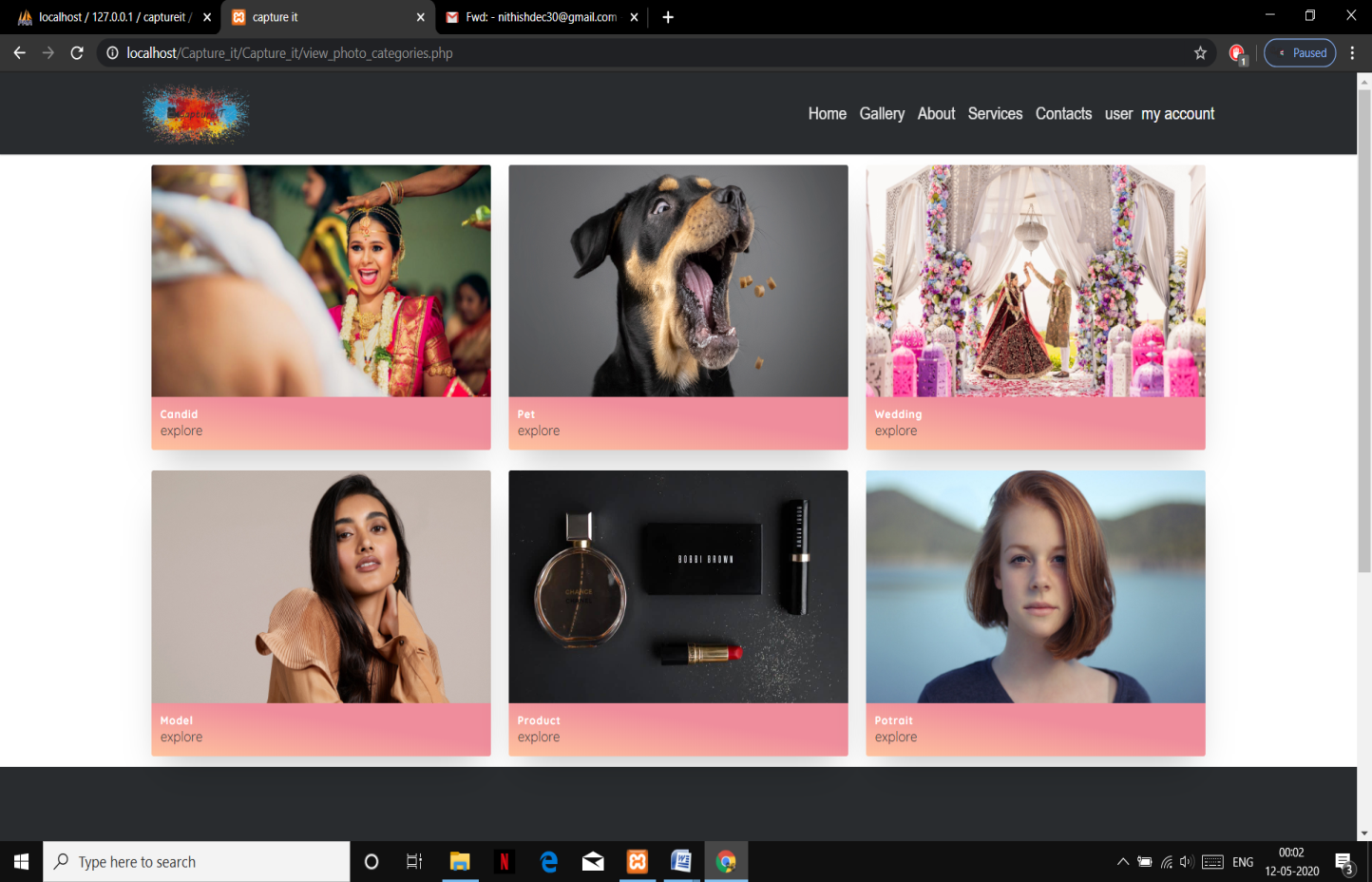
Forgot password



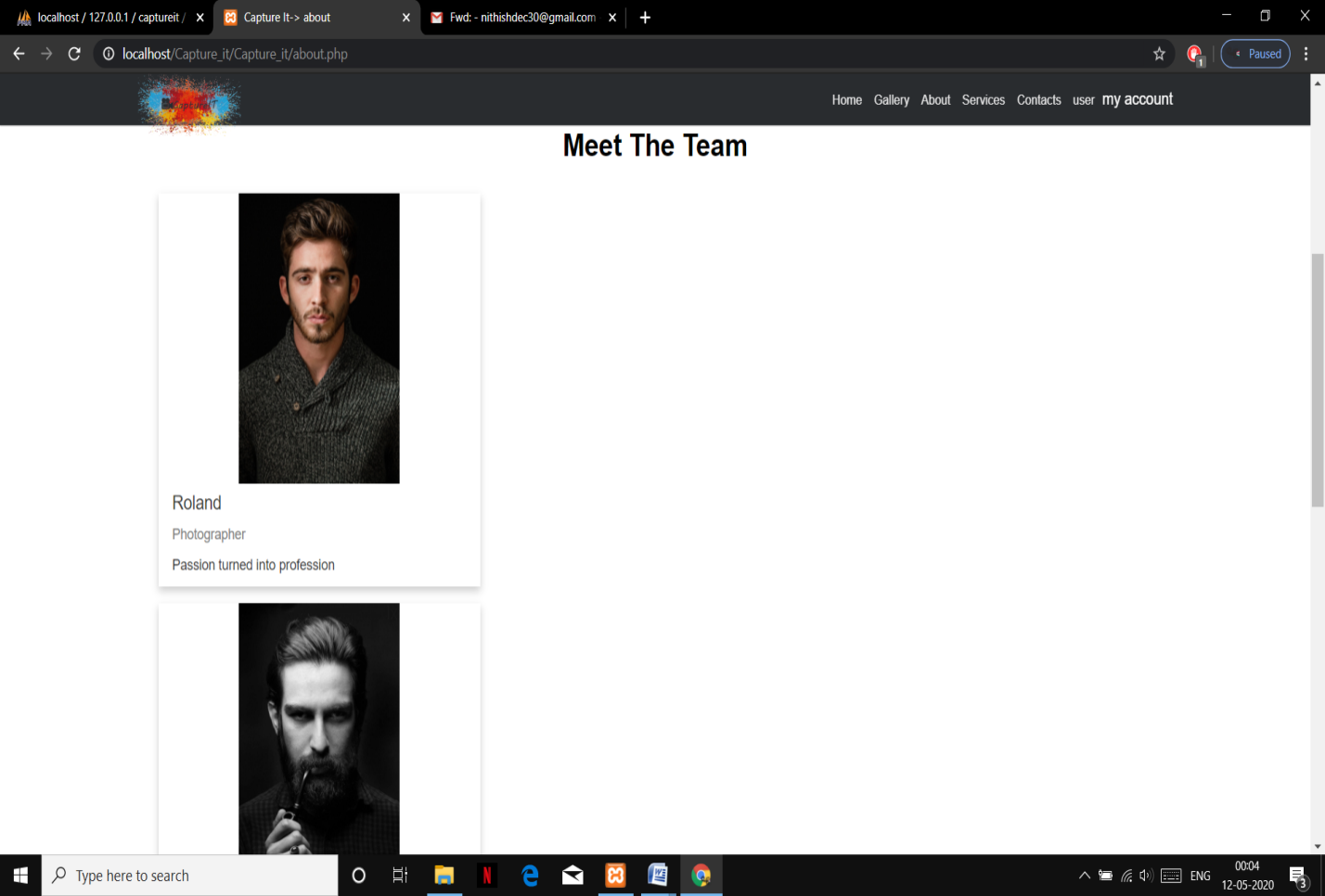
Gallery



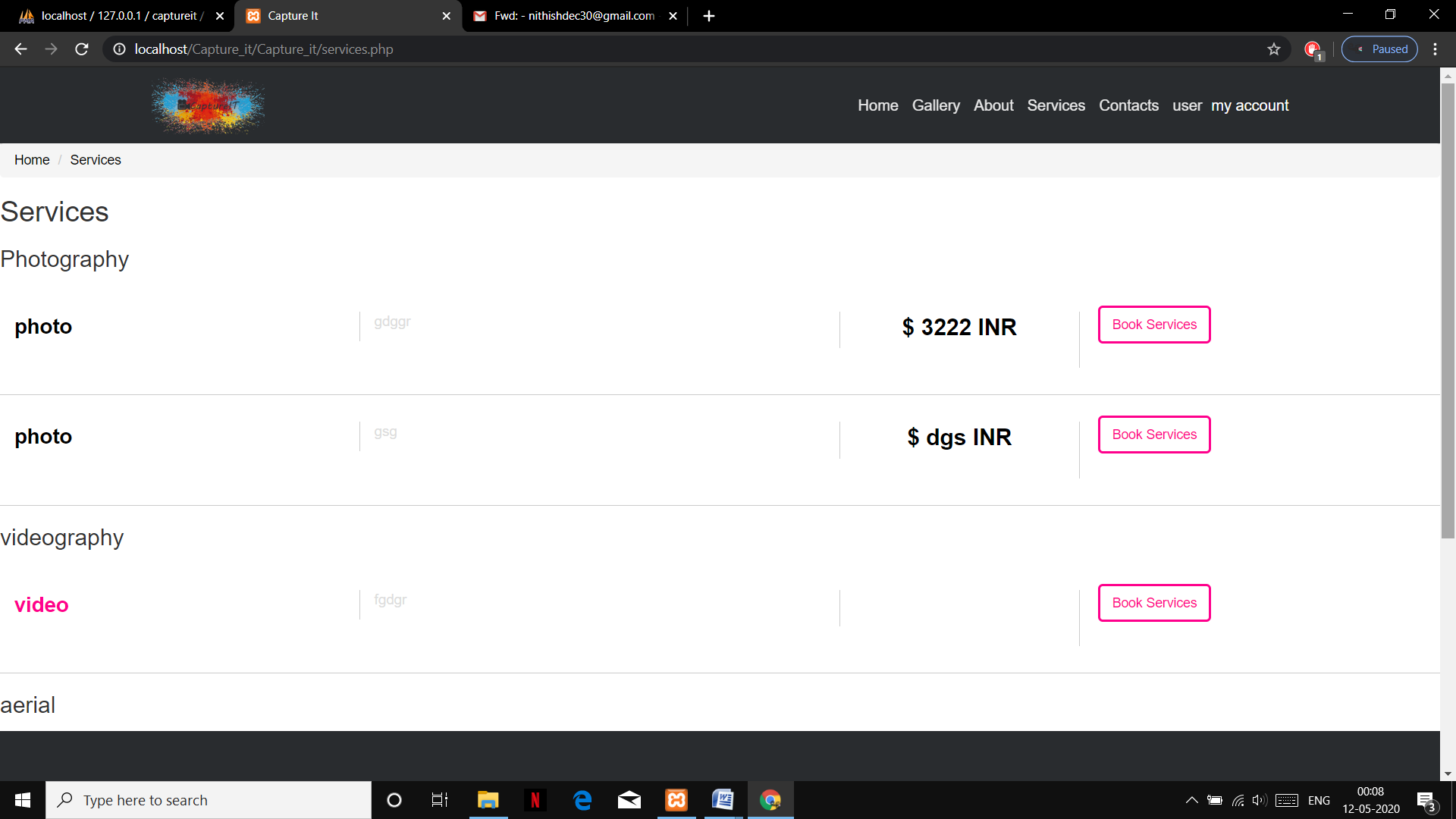
Categories



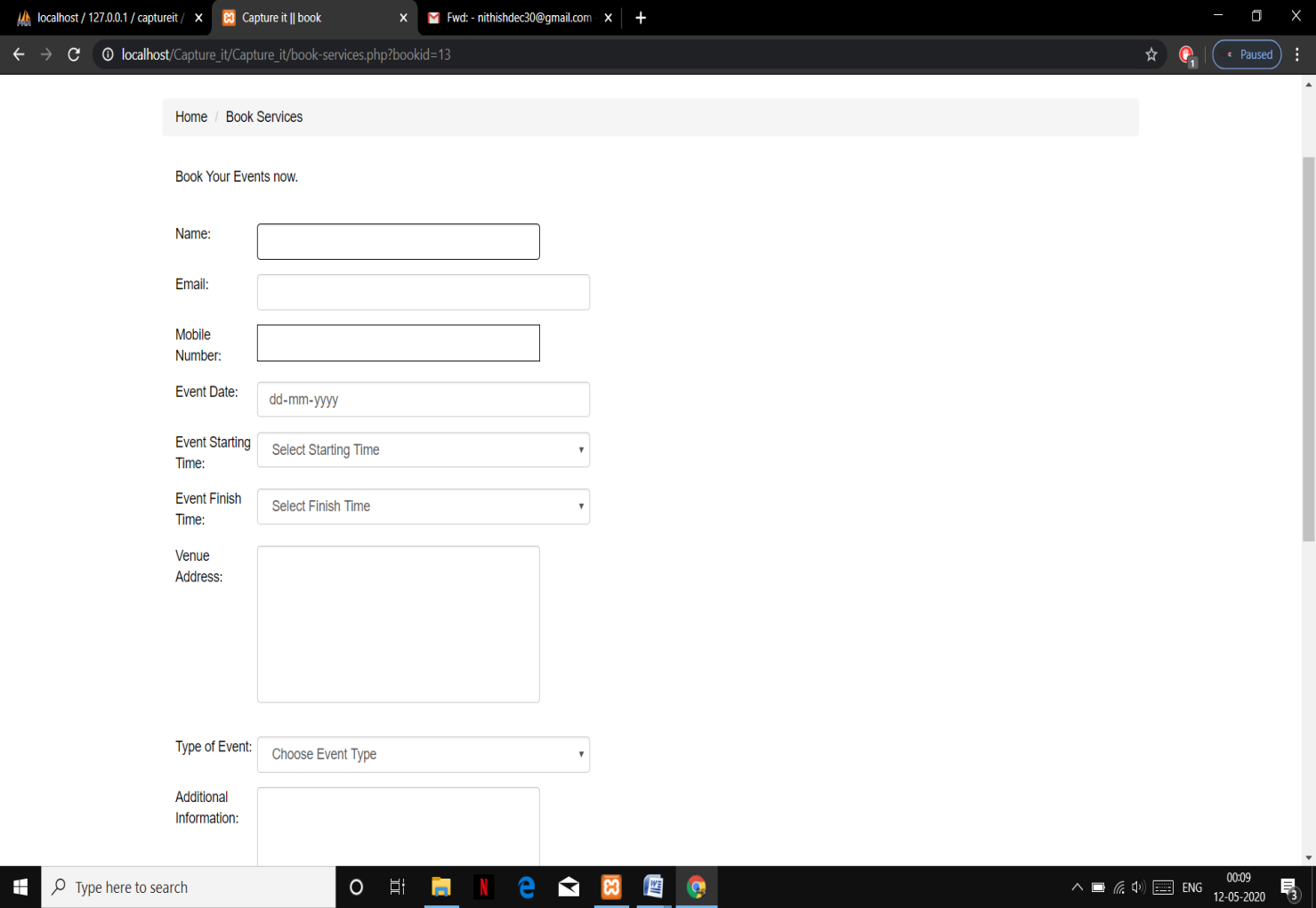
About

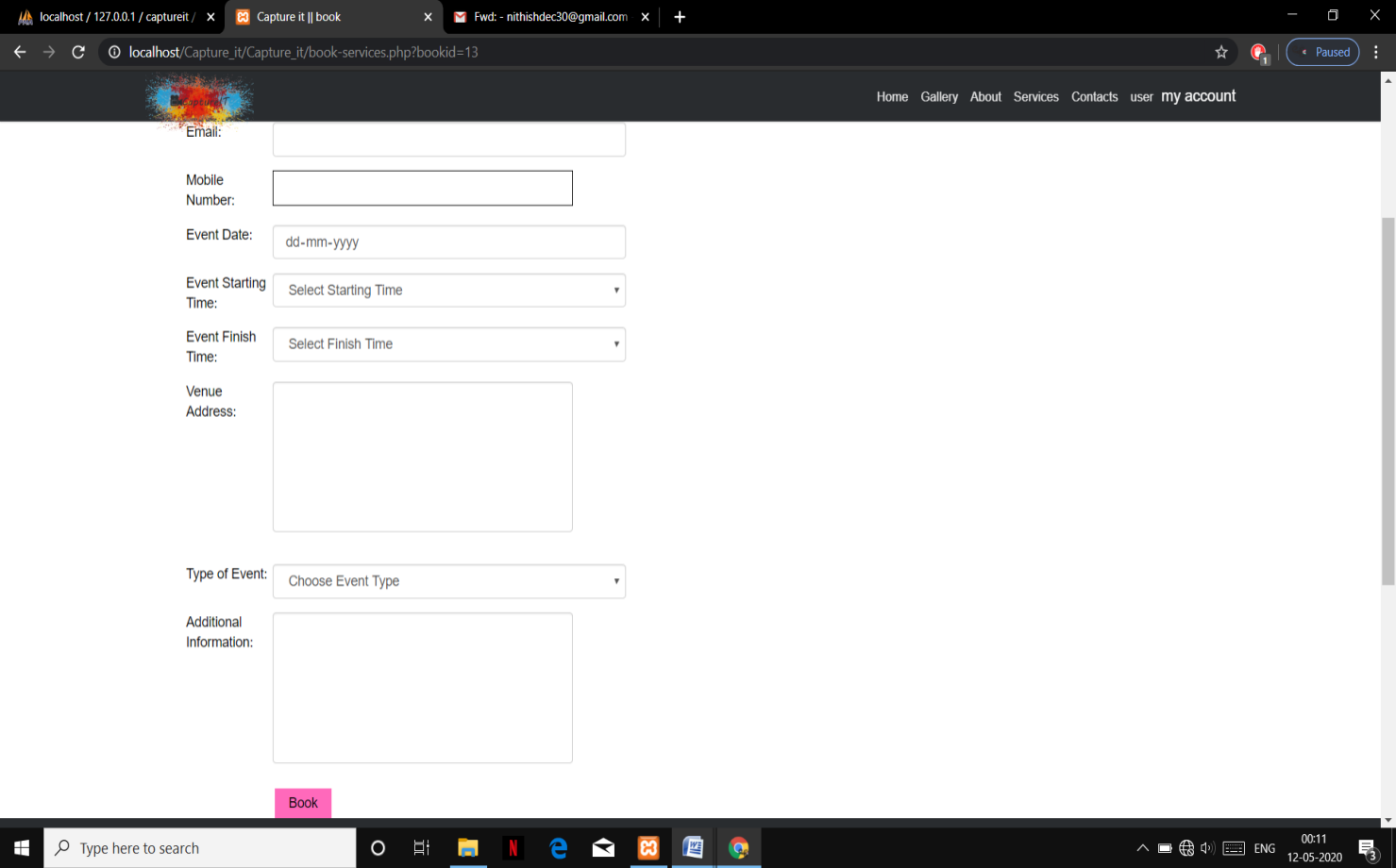


Services

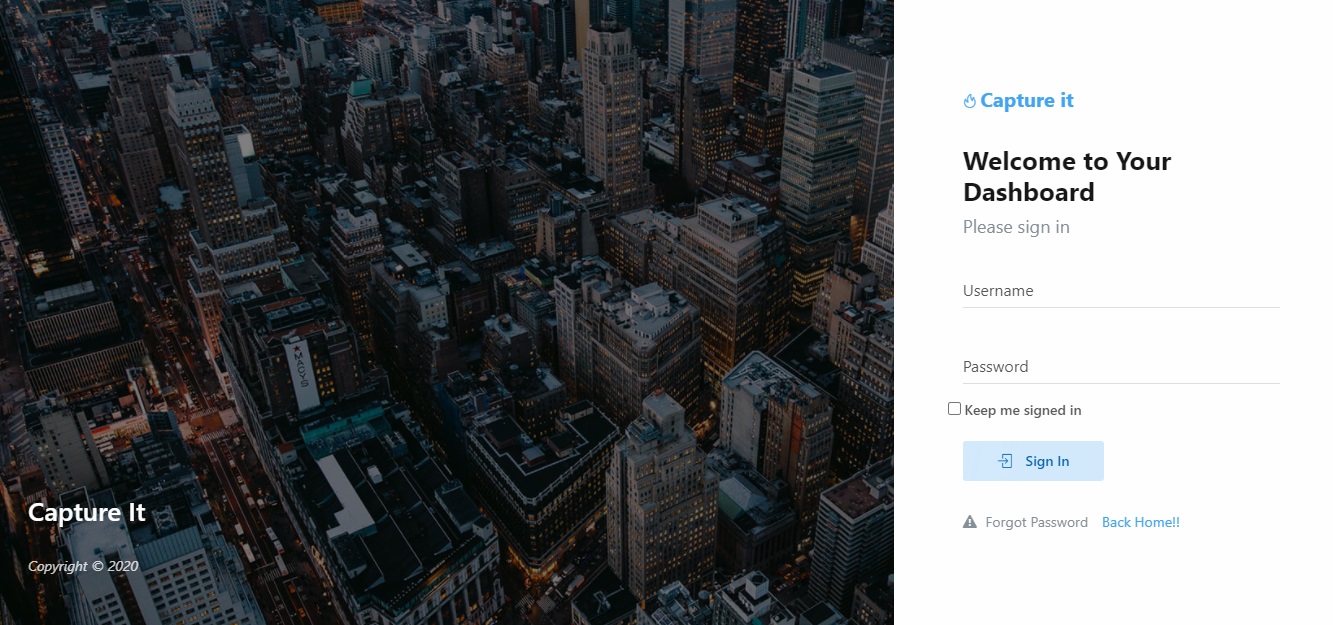


Book services

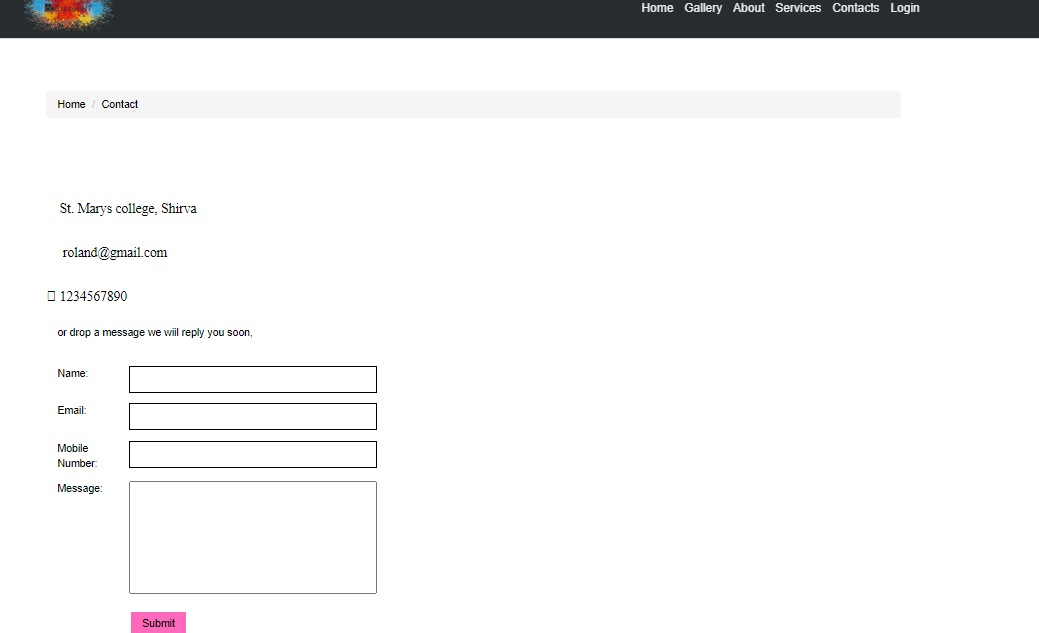




Admin Dashboard



Contact



My account

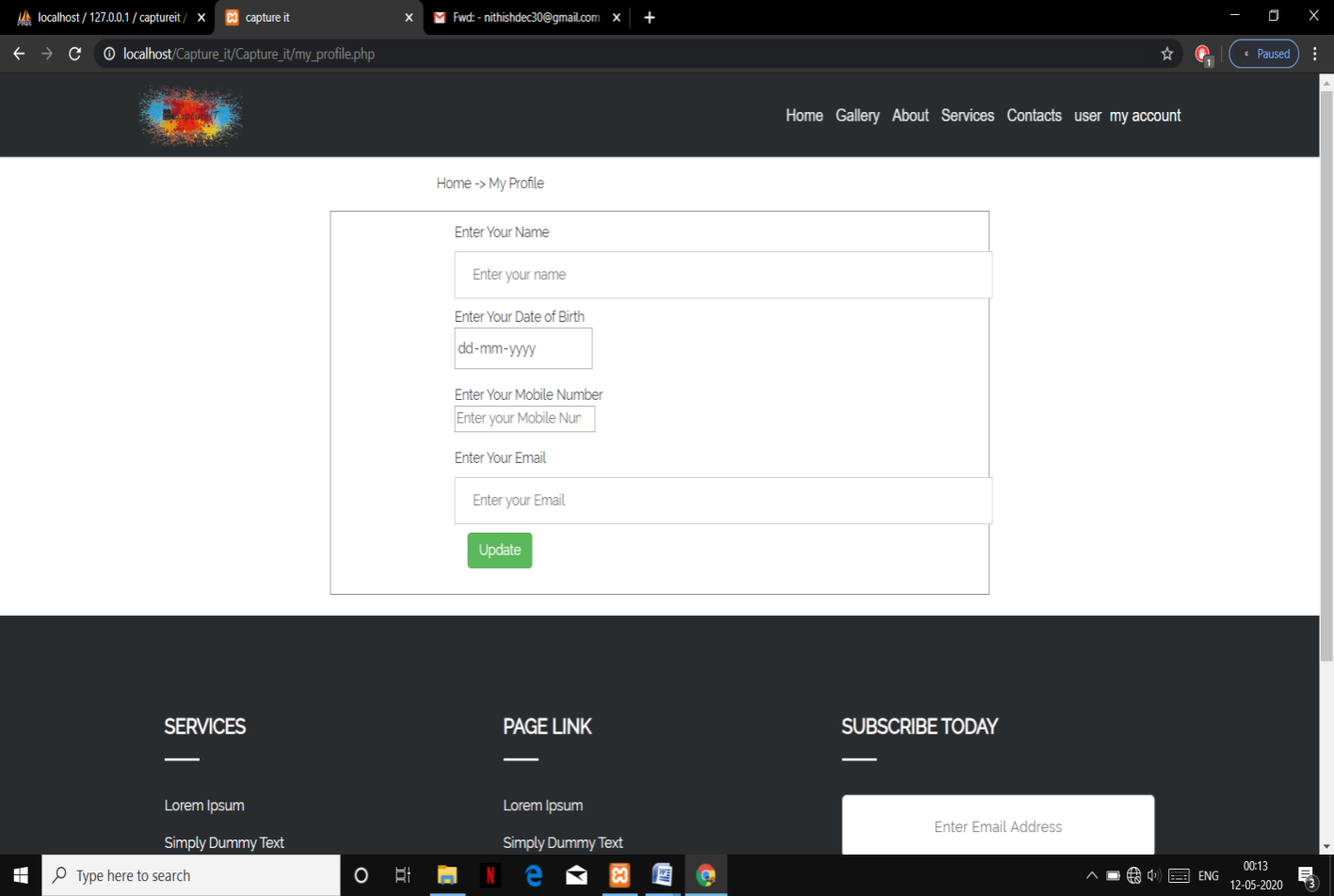
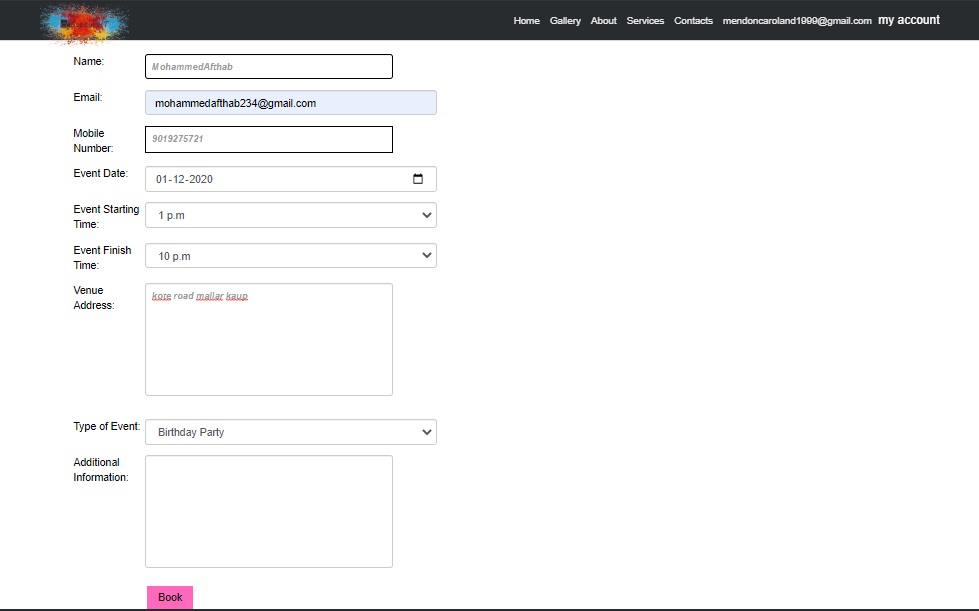
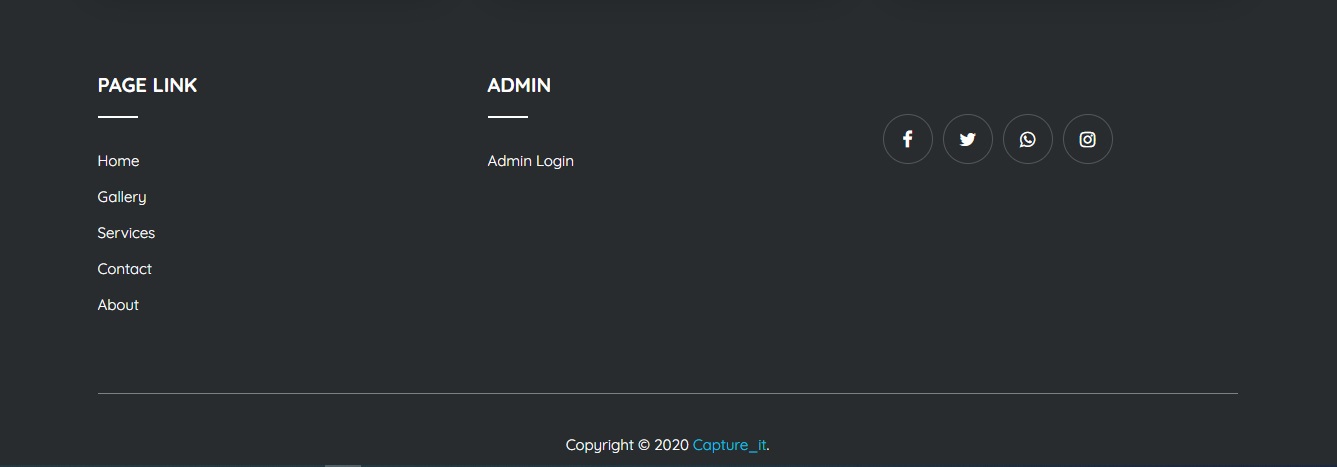


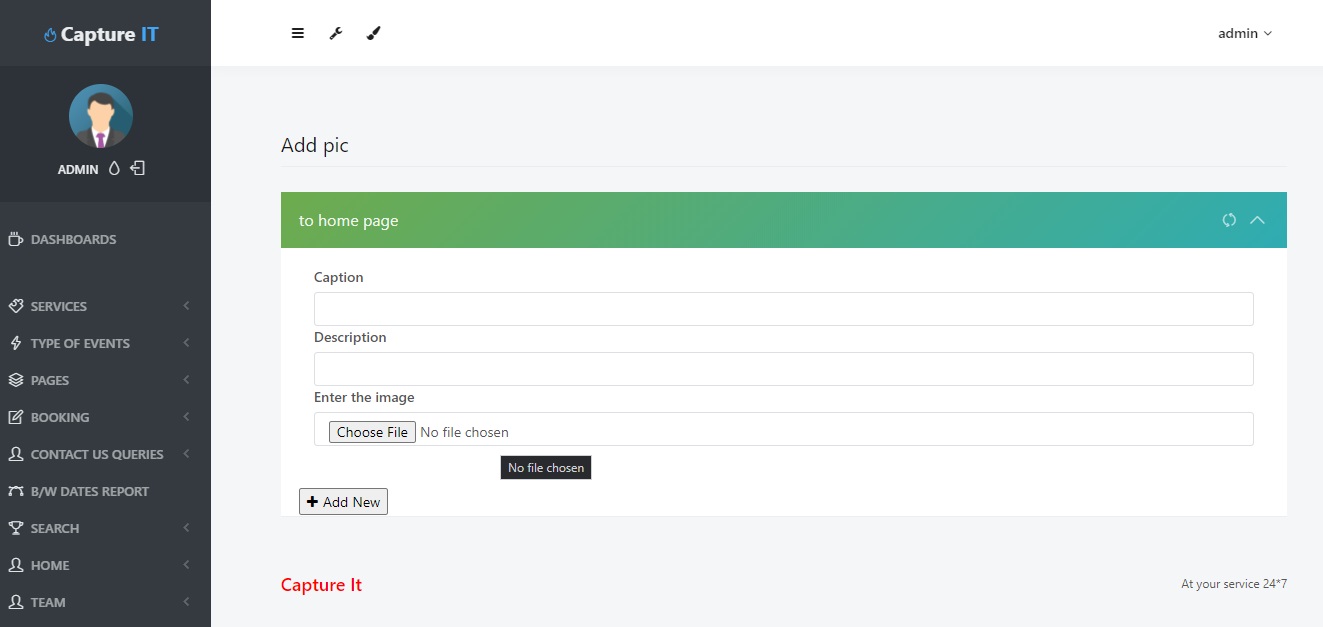
Photo Booking



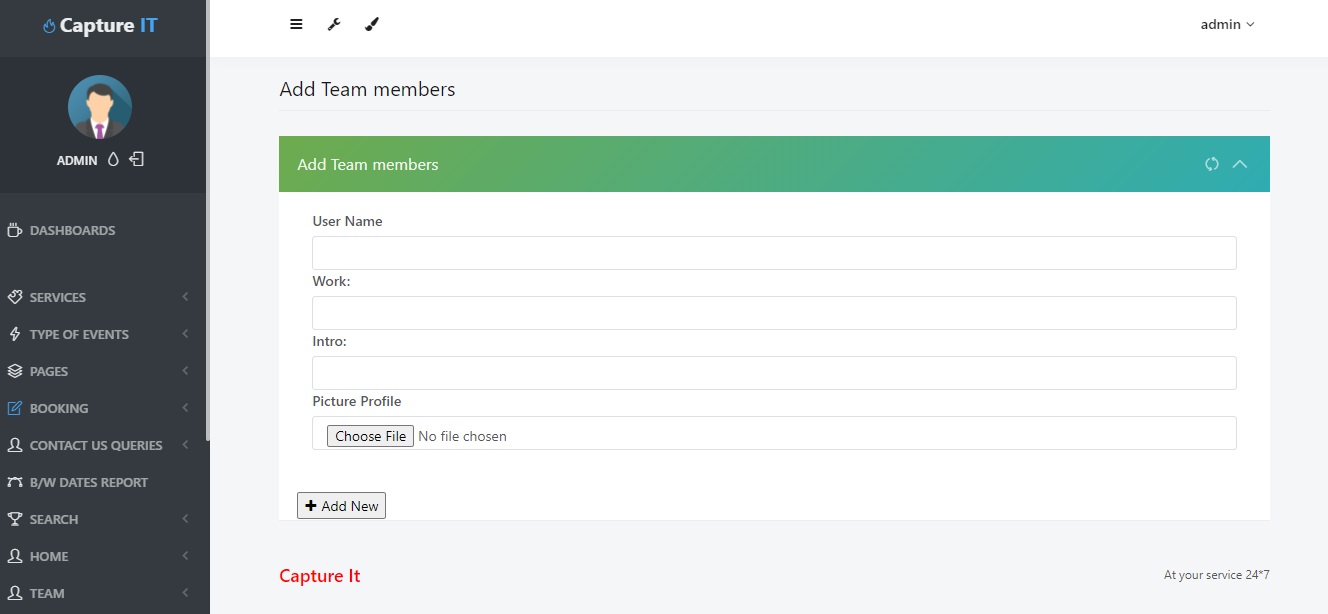
Footer



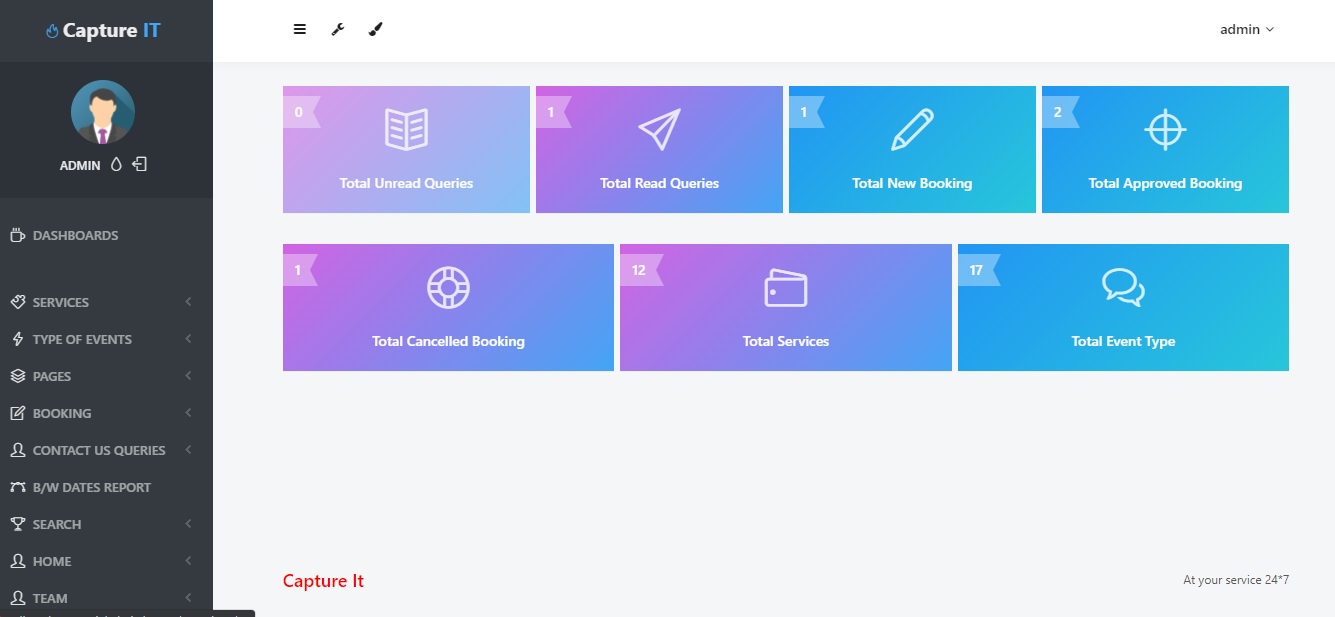
Add work



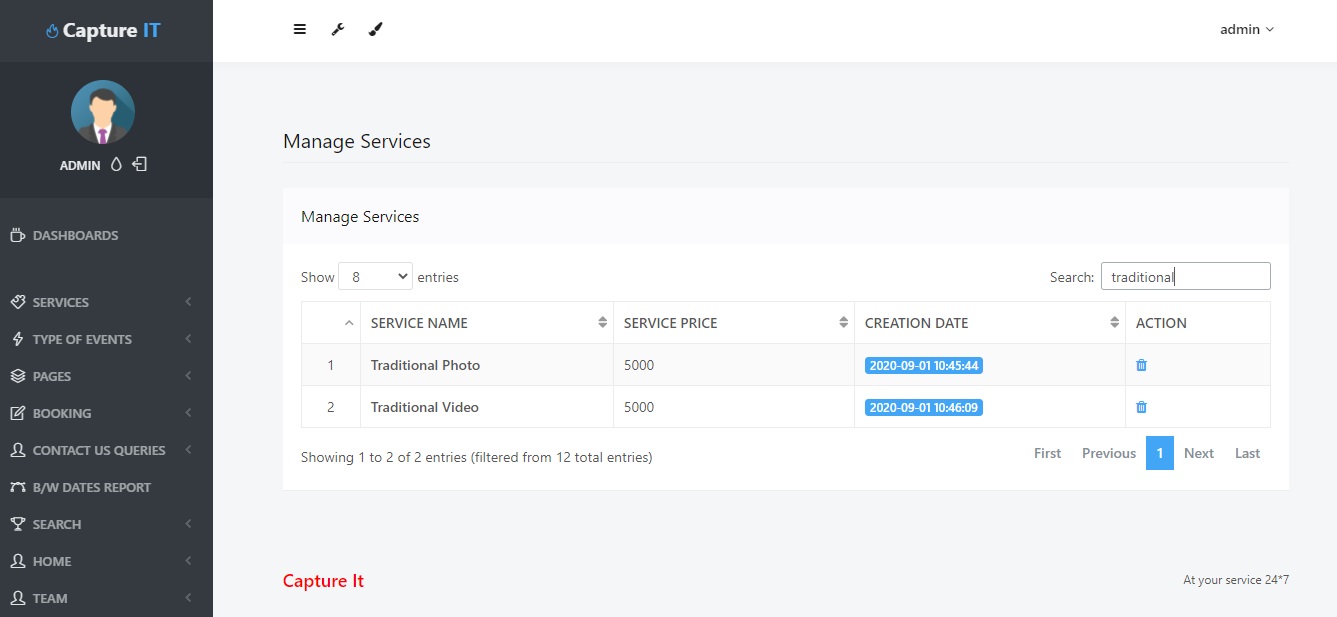
Add team



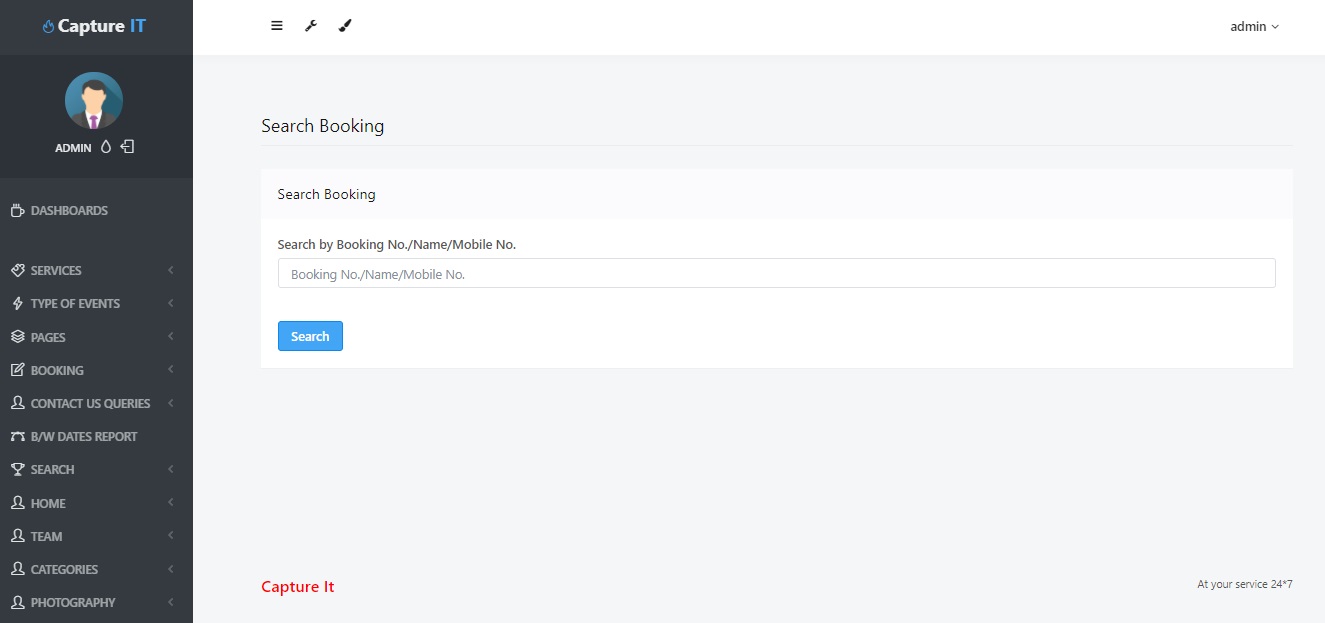
Admin Dashboard



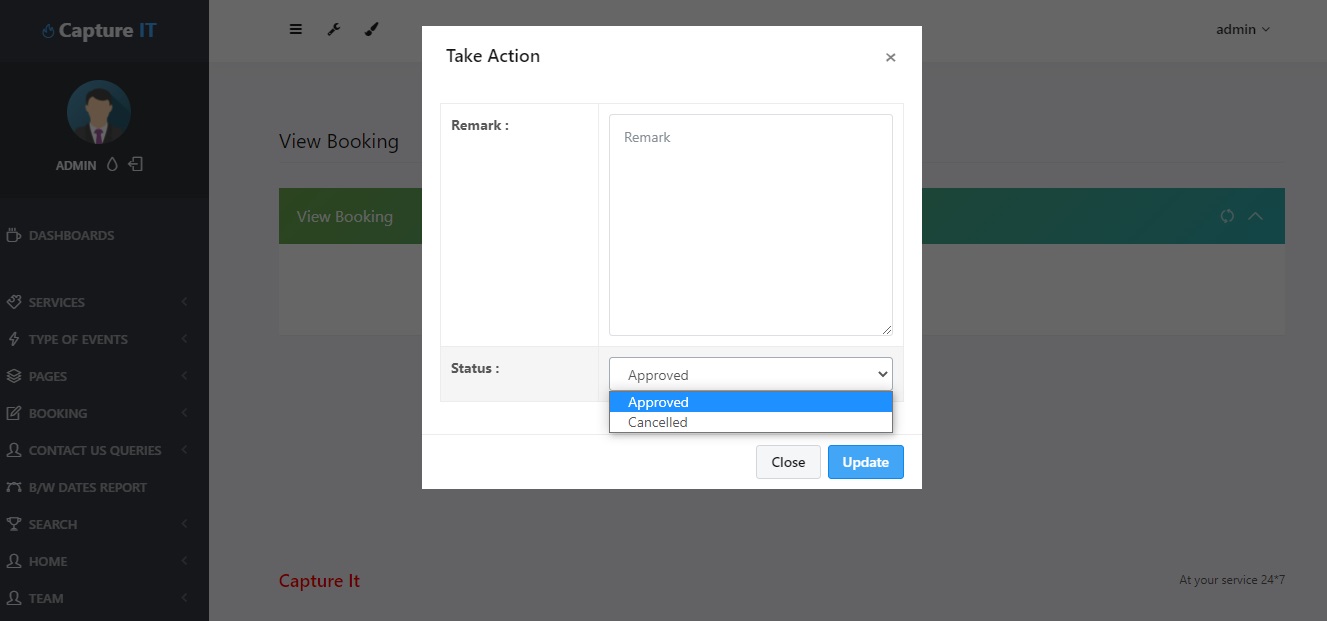
Search services



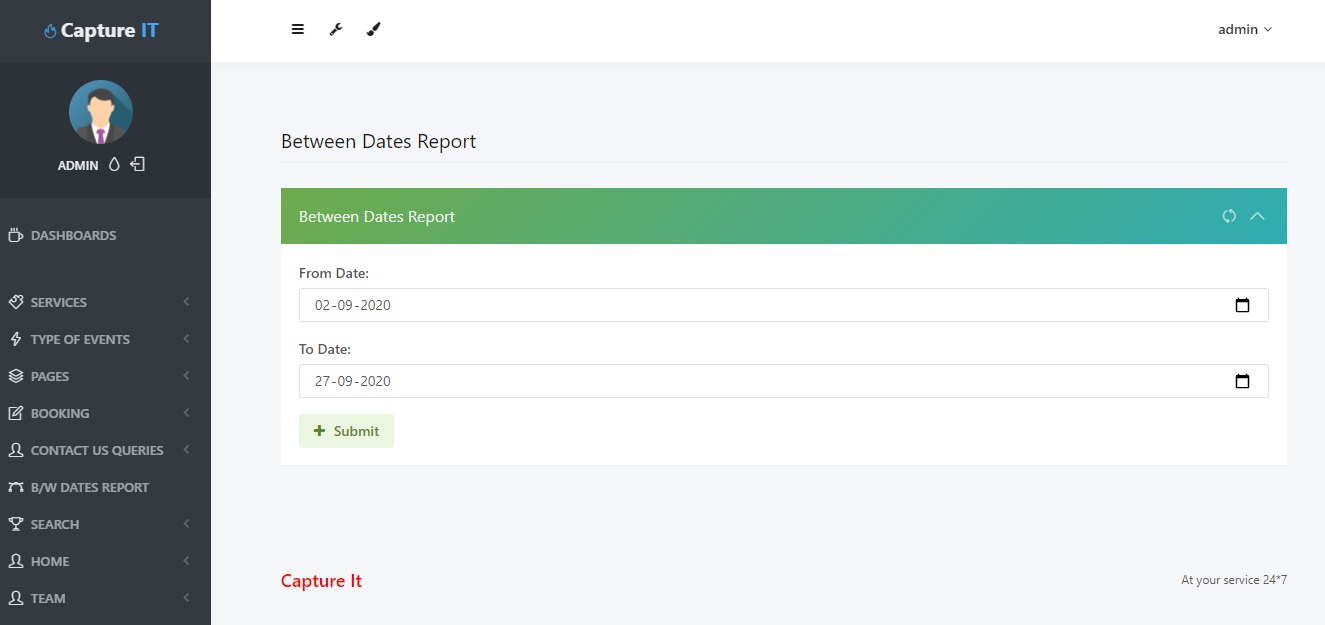
Search Book



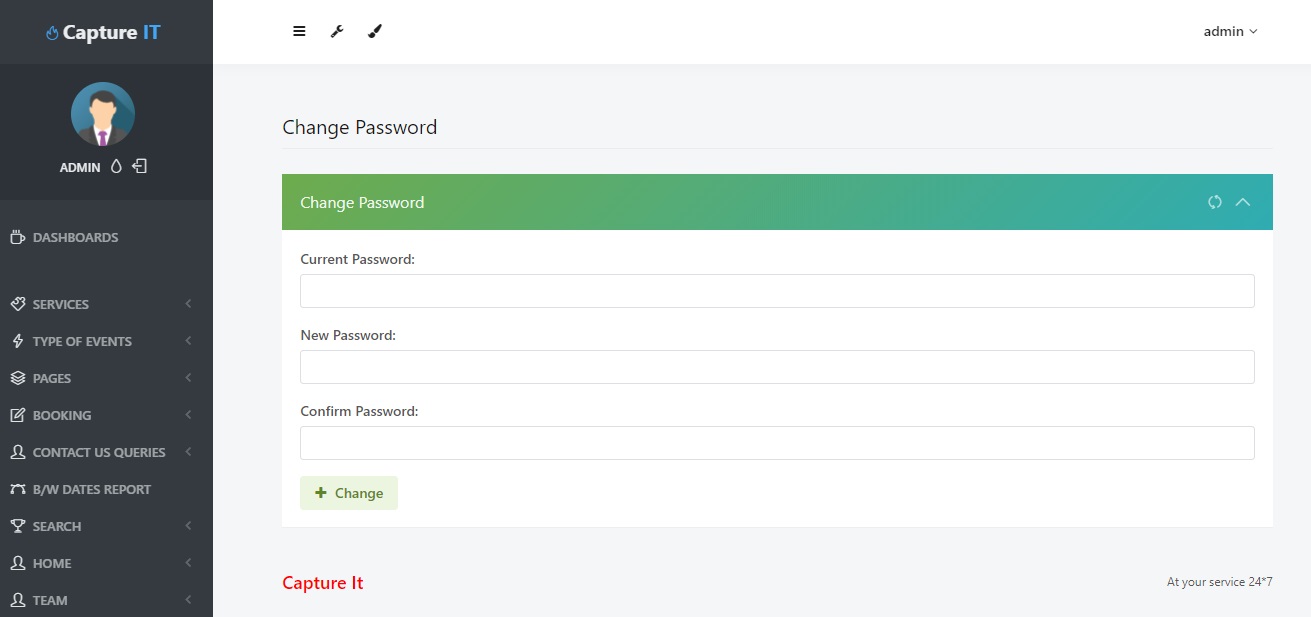
Take Action



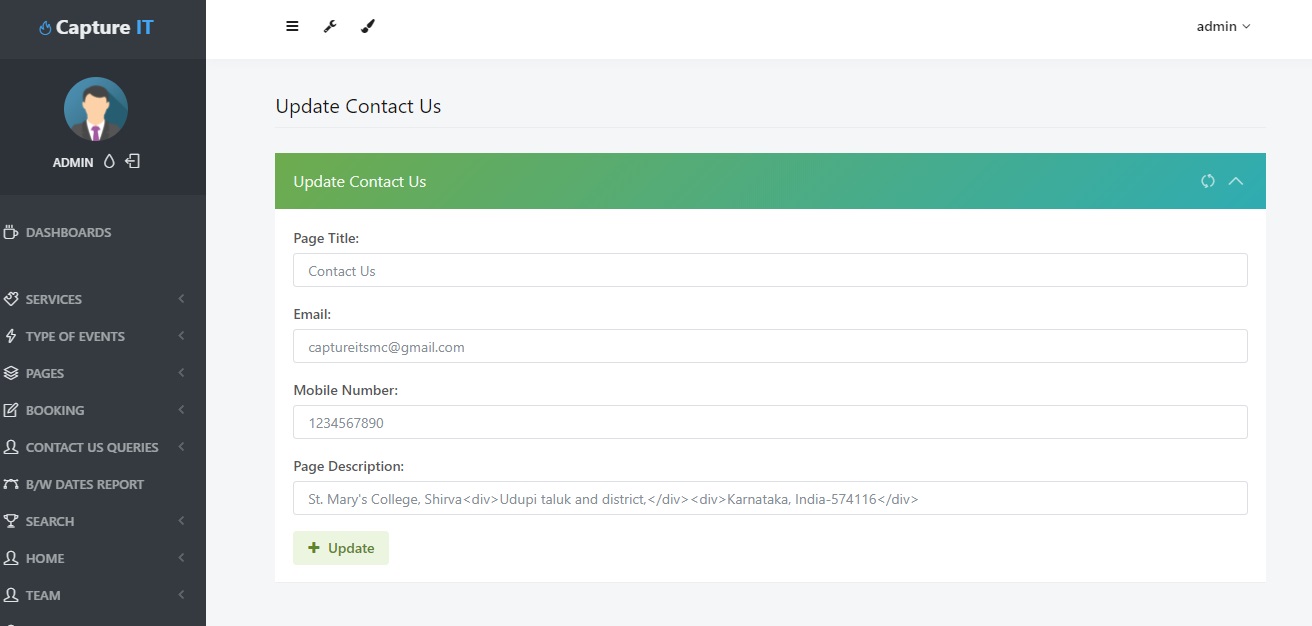
Date Reports



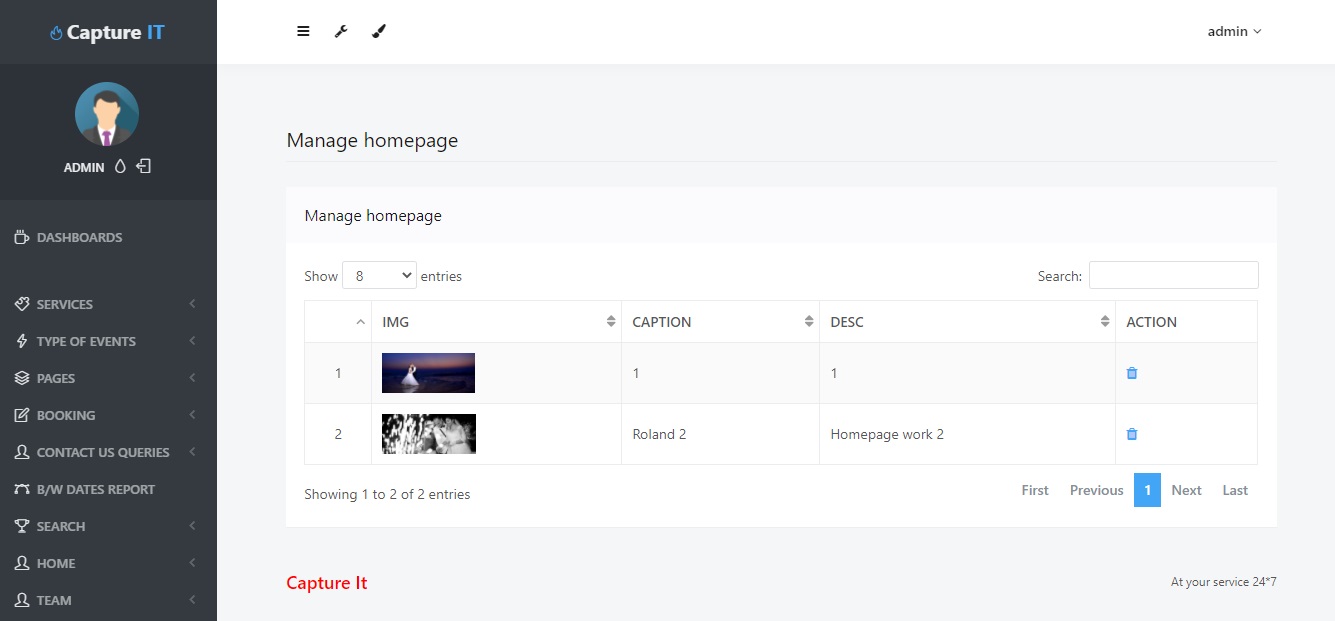
Change Password



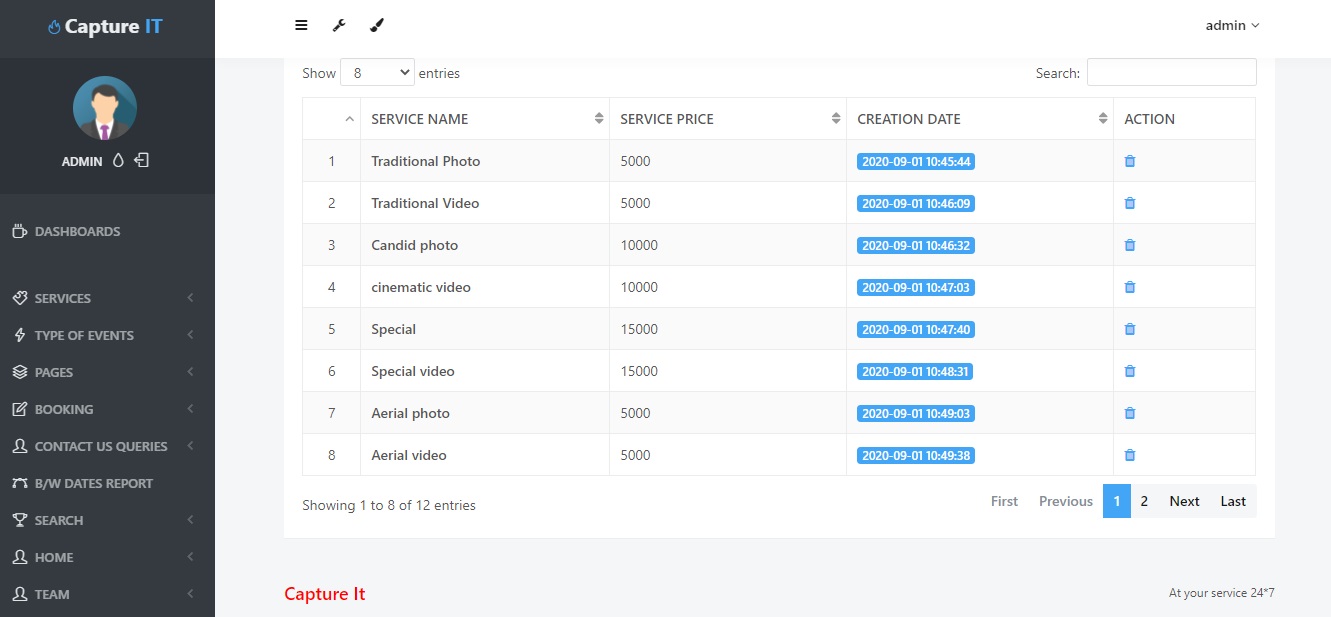
Contact Us



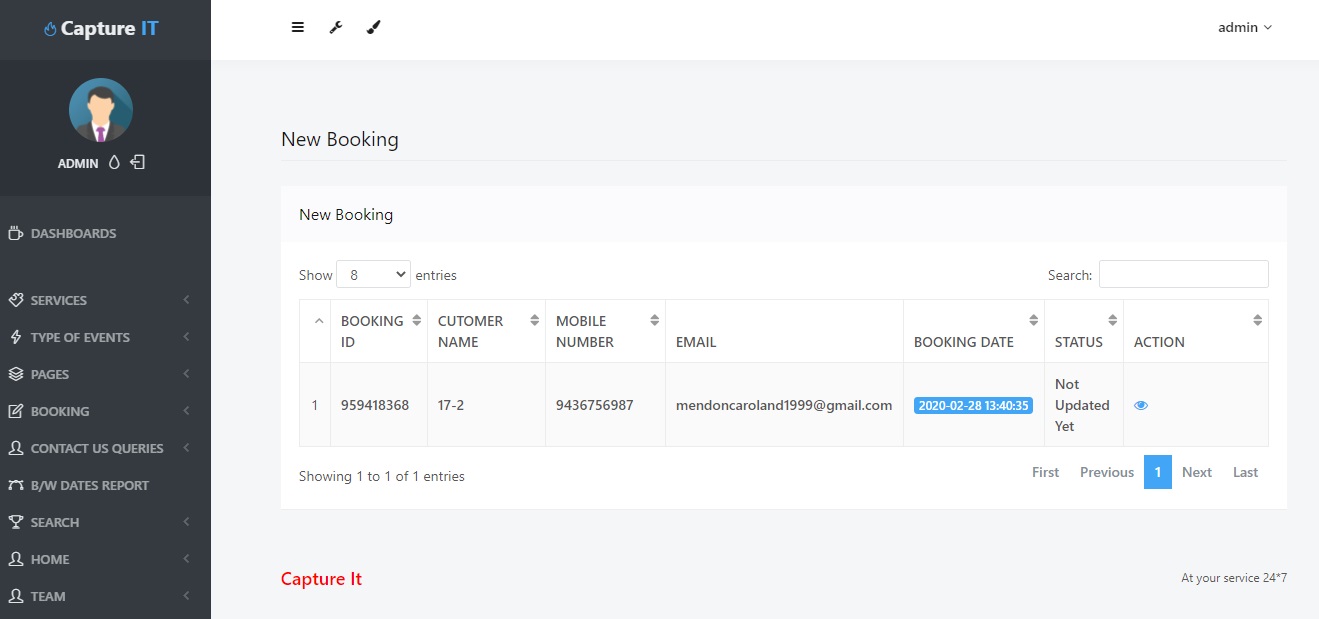
Manage Homepage



Manage Services



New Booking



Future scope

**Chapter 10**

**Future scope of the project**

* In future we will like to bring this project for smartphones.
* This project is more userfriendly by using biometric equipments.
* In future we will add more security features.
* This project is usefull to book the expert whom the user like.

Limitations

**Chapter 11**

**Limitations**

11.1 limitations

* slow internet connection may consume to load pages.
* Cannot book more than one service at simultaneously.
* Required high data connection speed.

Conclusion and future enhancement

Conclusion and future enhancement

**Chapter-11**

**Conclusion and future enhancement**

12.1 conclusion

“Captureit” is developed for the purpose of computerized online booking of services like photography,videography,aerial and its types. it is usefull in maintaining the photographer details.since it involves a lot of work in booking photography related service ,enquiry and maintaining their record manually.its very difficult for the user and admin ,as well as time consuming too.this software helps to enquire and book a photography service efficiently.

BIBLIOGRAPHY

**Chapter 13**

**BIBLIOGRAPHY**

12.1 Book reference

* Software engineering by Pankaj jalote
* HTML and CSS made simple by Ivan bayross
* Database management system Sudarshan prabhu

12.1.2 Web reference

* [www.w3schools.com](http://www.w3schools.com)
* [www.phptutorials.com](http://www.phptutorials.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)